(2)

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

2

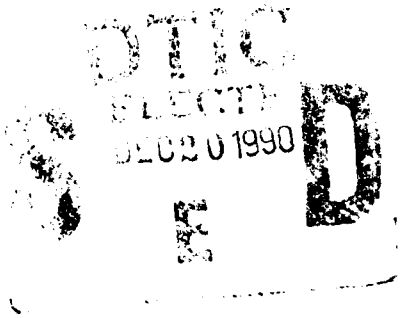
AFIT/GSM/LSQ/90S-14

DEVELOPING COMPUTER LITERACY AMONG INCOMING AIR FORCE INSTITUTE OF TECHNOLOGY GRADUATE STUDENTS

THESIS

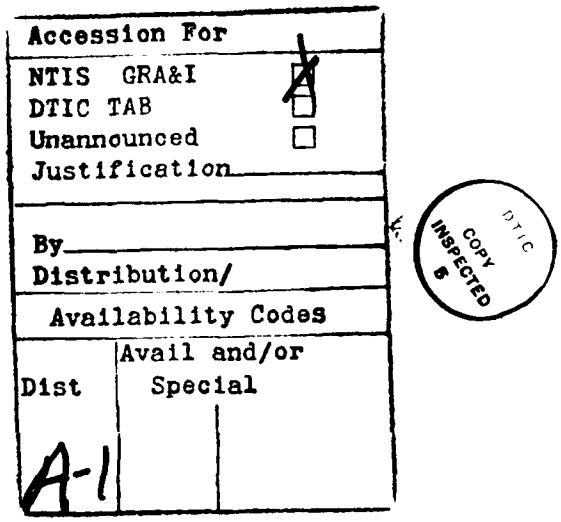
Stephen M. Heaps, Captain, USAF

AFIT/GSM/LSQ/90S-14



Approved for public release; distribution unlimited

The opinions and conclusions in this paper are those of the author and are not intended to represent the official position of the DOD, USAF, or any other government agency.



AFIT/GSM/LSQ/90S-14

DEVELOPING COMPUTER LITERACY AMONG INCOMING AIR FORCE INSTITUTE OF TECHNOLOGY GRADUATE STUDENTS

THESIS

Presented to the Faculty of the School of Systems and
Logistics of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Systems Management

Stephen M. Heaps, B.S.
Captain, USAF

September 1990

Preface

This research was intended to provide the faculty of the School of Systems and Logistics, Air Force Institute of Technology (AFIT) with a course designed to fulfill the need to introduce incoming graduate students to the applications software used at AFIT. By researching the current literature on learning theory and computer literacy, the author was able to develop a course that will promote minimum computer competency among the incoming graduate students. The revised QMGT 290, Introduction to AFIT Computers, course developed by this research was instituted by the AFIT faculty starting with the 91S/D class at AFIT. This research was borne out of my dismay with the lack of applications software instruction in the former QMGT 290 course. Since many of my classmates did not possess the applications software knowledge necessary to effectively tackle the course work at AFIT, I felt this research was worthwhile. My sincere thanks to my advisor, Lt Col Richard Peschke, PhD, for driving my energy toward a worthwhile goal and also for his patience in the editing phase of this research as well as the VAX VMS manual I created with his guidance. In addition thanks goes to Mr James Bacus, Assistant Director of the Office of Computing Activities, University of Dayton for their help in the early phases of the research. Thanks to Ms Charlotte Fordyce, Course Director of CS-205, Computer Literacy for Business Majors, Wright State University, for allowing me to observe her class to gather ideas to aid in structuring a course of this type. Thanks to Major Jake Simons, Major Maurice Riggins, Major Rich Kern, Lieutenant Don McNeeley (USN), Professor Dan Reynolds, Professor Jim Meadows and countless others who helped keep me focused toward a successful completion of this worthy cause. Lastly, and mostly, thanks to my biggest fan, supporter, and sounding board, my wife, Lorrie. Her patience and understanding kept me going.

Stephen M. Heaps

Table of Contents

List of Figures

Abstract

The purpose of this research was to develop an introductory level, applications software oriented, computer course for the School of Systems and Logistics, Air Force Institute of Technology (AFIT). This course would be used to introduce new graduate students to the applications software used in the AFIT graduate program. The research was designed to answer four basic research questions:

1. What computer skills are necessary for minimum computer literacy at AFIT?
2. Which application software packages are used in the AFIT graduate program?
3. Given the course-time available to teach QMGT 290, what should be taught to maximize the literacy of the incoming students?
4. What teaching method(s) is/are best to teach computer applications? The research began to address these questions with a review of current literature on learning theory and computer education. From this body of literature the author extracted a model for developing computer curriculum based on a three levels of proficiency. This model was used to develop the revised QMGT 290, Introduction to AFIT Computers. The revised course was developed using the familiar building block approach from learning theory. The literature review was further used to determine the most practical approach to teaching the course. The revised course contained a combination of lecture, hands-on, and take-home problems and exercises. The objective of the revised QMGT 290 course was to ensure a minimum competency level among all new students so that students possess the knowledge necessary to use the computer tools provided at AFIT.

Over 75% of the students completing the revised course agreed that the course met its primary objective. The most common complaint about the revised course was that there was too much material presented for the time allotted to class. This resulted in cursory exposure to the software instead of in-depth knowledge. As a result of the comments from both the students and feedback from the instructors, recommendations were made to modify the course to expand the in-class time and investigate the possibility of grouping the sections according to proficiency level. Such grouping will allow structuring the course to address the needs of the students. A further recommendation was made to continue the application orientation of the course in the future.

DEVELOPING COMPUTER LITERACY AMONG INCOMING AIR FORCE INSTITUTE OF TECHNOLOGY GRADUATE STUDENTS

I. Introduction

General Issue

Computers abound in every aspect of society today, including education. Every year, student success in higher education becomes more dependent on applying computer skills to solving problems. The Air Force Institute of Technology (AFIT) demands computer literacy from its graduate students to assure competent use of the electronic resources provided in the classroom.

Specific Problem

The faculty should develop an introductory course that promotes basic competency in the major software application environments used in the graduate curricula. Currently, the AFIT faculty offer a primer course to introduce students to the AFIT computer systems. The course objective is to provide exposure to some of the basics of microcomputer use so that students can better accomplish academic assignments using the computer. The problem is that QMGT 290, "Introduction to AFIT Computers," does not currently provide exposure to the major application

software used during programs the graduate program and therefore does not fully meet the objective of computer literacy.

Tentative Hypothesis and Research Questions

This research provides a recommendation to the faculty for changes to QMGT 290. The author proposes a new course intended to build basic computer literacy skills in the major software applications environments that are currently in use at AFIT. As a basis to build this course the researcher addressed the following Research Questions:

1. What computer skills are necessary for minimum computer literacy at AFIT?
2. Which application software packages are used in the AFIT graduate program?
3. Given the course-time available to teach QMGT 290, what should be taught to maximize the literacy of the incoming students?
4. What teaching method(s) is/are best to teach computer applications?

Scope of Problem

This research is limited to information gathered with respect to computer literacy levels for Class 91S/D of the AFIT School of Systems and Logistics. This research does not address the AFIT School of Engineering since Computer Literacy requirements vary greatly between the two schools.

II. Literature Review

Chapter Overview

This chapter starts with a definition of computer literacy followed by a review of the literature in professional and trade journals dealing with developing computer literacy. Researching the literature for learning theory and computer literacy provides insight into methods and practices of computer literacy training which will aid in proposing a curriculum necessary to develop basic computer literacy among AFIT students. Computer literacy training follows the same basic principles as any other learning experience; studying learning theory helps one understand basic concepts of the learning paradigm. These concepts can be honed to apply to computer education as well as general education. Since the extensive flood of microcomputers into society, computer training and computer literacy have been the focus of national, state, and local attention (2:61; 20:232; 21:42; 23:12). There is a wealth of information available to help develop curricula for this area (20:232; 21:42; 23:12; 24:21). This search is limited to a sample of professional, trade, and popular journals, and books. The author used a sample of the available literature because the magnitude of information relevant to this topic precludes an exhaustive search in the time allotted. There are thousands of articles that address computer literacy and computer training. A complete search would be impractical. In addition, the author discovered that although there is a great deal of information printed on the subject, much of the information has a common focus. This limitation of scope does not reduce the validity of the research since generally speaking, most of the literature draws the same conclusions. These conclusions being that computer literacy is still the exception and

not the rule in America. The author researched literature on general learning principles first to build a foundation for learning principles and then proceeded to research the specific topic of computer training. This chronology of research is appropriate in that the basics of learning theory are the same regardless of whether one is learning computers or some other subject such as mathematics. In fact, it would be inappropriate to discuss computer learning without a learning theory foundation firmly in place as the fundamental basis for discussion.

Introduction

Computers have entered every facet of one's life. A person cannot go to a grocery store, get money from an automated bank teller machine, register an automobile, fly in an airliner, or even, in some instances, get a paycheck without a computer. This is the computer revolution. With computers used in most avenues of life, it is becoming more important to have some basic computer literacy skills to function effectively in society today. Charles Witten wrote in his Naval Postgraduate School thesis, "failure to learn about computers and their use will leave the uninitiated in a position of being functionally illiterate, without the skills needed to exist in an information society." (25:8)

Basic computer literacy for everyone is justified by the intensive use of computer technology in our modern world. Witten points out that basic computer literacy by no means is intended to qualify a person for a job as a program analyst or programmer. (25:8) Instead, it is aimed at making the population aware of computers and their uses.

Computer Literacy

A Definition. What is computer literacy? Chances are that one could ask that question to ten different people and get ten different answers. The computer revolution has not provided a clear-cut definition. Even the authorities do not agree. Definitions range from being aware of the capabilities of computers (23:10) to being able to program in some sophisticated language (20:232). There is so much confusion that when the State of Kentucky tried to define computer literacy in legislation to mandate computer literacy training in the public schools, the lawmakers gave up and left the definition up to the individual school districts (21:42). As a result, the State of Kentucky has about 180 different approaches to computer literacy for 180 different schools districts (21:40). Because of this problem, the Committee on Computer Education of the Conference Board of the Mathematical Sciences recommended financial support from the National Science Foundation to develop at least one standard secondary school course on computer literacy (23:10). Troutner proposed that the course discuss how computers are used, the impact on society of the computer, the capabilities and limitations of computers, and introduce the idea of algorithms as they apply to flowcharting and programming (23:10). It seems that this course and the definition of computer literacy that was to result from it, never materialized. A major problem with such an undertaking is that because of the very nature of the computer industry, an industry seeing revolutionary developments almost daily, technology can not stabilize long enough to provide a definition that will live past the time that it is agreed upon (21:42). Michael Simonson, et. al., in their article "Development of a Standardized Test of Computer Literacy and Computer Anxiety Index," say the term falls within three overlapping approaches. The first approach

equates computer literacy to being able to program. The second approach equates computer literacy to possessing general knowledge of computer uses and operations, to include an understanding of capabilities, applications, and algorithms. The third approach is a combination of the first two (20:232). Simonson, et. al. developed a much more comprehensive definition:

> Computer literacy is defined as an understanding of computer characteristics, capabilities, and applications, as well as an ability to implement this knowledge in the skillful, productive use of computer applications suitable to individual roles in society (20:233).

Simonson, et. al. stated that knowledge and skills were divided into four groups: computer attitudes, computer applications, computer systems, and computer programming.

They further defined these four groups as follows:

> COMPUTER ATTITUDES refers to an individual's feeling about the personal and societal use of computers in appropriate ways. Positive attitudes include anxiety-free willingness or desire to use the computer, confidence in one's ability to use the computer, and a sense of computer responsibility.
>
> COMPUTER APPLICATIONS refers to the ability to responsibly evaluate, select, and implement a variety of practical computer applications to do meaningful and efficient work based on an understanding of general types of applications, capabilities, and limitations of types of applications, and societal impact of specific applications.
>
> COMPUTER SYSTEMS refers to the appropriate, knowledgeable use of equipment (hardware) and programs (software) necessary for computer applications.
>
> COMPUTER PROGRAMMING refers to the ability to direct the operation of the computer through the skillful use of programming languages. This would require an understanding of problem solving strategies, algorithms, flowcharts, languages, and programming. (20:234)

Lieutenant Colonel Richard Peschke, course director for the "Introduction to AFIT Computers Systems," defines computer literacy as the ability to understand and use computer systems to accomplish applications-oriented tasks without computer phobias (19). For the purposes of this research, the author defines computer phobia as an abnormal fear of computers. The author has adapted Lieutenant Colonel Peschke's definition of computer literacy for this research.

Why Teach Computer Literacy. Why is computer literacy important? Probably the easiest answer to that question was proposed in Charles Witten's master's thesis, "Development of IS2100, An Information Systems Laboratory." Witten said, "failure to learn about computers and their use will leave the uninitiated in a position of being functionally illiterate, without the skills needed to exist in an information society" (25:8). The language of computers is a separate and complex language. If a person elects not to become computer literate, technology will leave that person behind. As more and more jobs become computer oriented, lack of computer competency will jeopardize one's ability to find employment in tomorrow's industries. This alone should be reason enough to learn about computers.

The Air Force is taking computer literacy seriously. In a report entitled "Computer Training/Literacy/Proficiency," the Air Force Management Analysis Group (AFMAG) on Data Systems Management and Manpower recommended to the Air Force Chief of Staff that computer literacy training be required for all Air Force managers. The overall finding of the report was that many Air Force managers are not sufficiently knowledgeable about computers and thus may be unable to manage these Air Force resources. The report recommended that a computer short course be developed and conducted throughout the service. Air University was tasked to develop

appropriate computer literacy courses for Squadron Officers School, Air Command and Staff College and Air War College (9:10). Air Training Command was further tasked to develop an easy to teach computer literacy course that could be taught at base level and to Officer Training School (OTS) and Reserve Officer Training Corps (ROTC) students (9:10). In addition, it recommended that professional military education include mandatory computer literacy training. (9:10) This author was not able to locate this course. It appears that the direction has not yet been implemented. The requirement did not disappear. In fact students at AFIT need to be computer literate to survive the graduate program.

The next important question is "Why is it so important that AFIT students either arrive computer literate or become computer literate within the initial four week short term at AFIT?" The answer is obvious if one looks at the requirements for many courses taught during the first quarter of graduate studies; the computer is a must at AFIT. The computer is used for Electronic Mail (many instructors give class assignments using this mode), most classes require a term paper, which could more easily be accomplished using a word processor, and early mathematics classes require extensive use of applications programs such as Mathcad™, a product of Mathsoft Corporation; Quattro Pro™, a product of Borland International; and Statistix™, a product of NH Analytical Software.

Thomas Spain comments that currently, most students leaving high school are not computer literate. He postulates, however, that in about ten years computer literacy courses will not be required in the colleges because students will enter college computer literate (21:42). He believes that even though it is hard to come to a consensus on what should be taught, most school districts are developing their own

guidelines instead of waiting for the states to make up there mind. As such many high schools and even some middle schools in America today have started to develop computer literacy training programs for their students (21:42). Until these courses are provided as a rule instead of as an exception, colleges, including AFIT, should provide an educational opportunity for students to become computer literate.

Computer literacy training is currently being taught through many avenues in America today. One need only open any Adult Education pamphlet or college catalog and see computer education courses. Advertisements like "Learn LOTUS 123 in 8 hours," "The Computer Advantage," "We turn PC Owners into PC Users," abound. So what is so hard about training people? To understand the answer to this question, one must understand basic education theory.

Learning Theory

What is Learning? According to most researchers, learning is the ability to do something new (12:36; 6:39). Researchers know that learning is a progressive process, a building up of basic concepts (6:39). A person striving to learn something new must move through stages from the simple, through the intermediate, to the advanced. If a person jumps into a complex stage without first successfully completing the simple and intermediate, that person will be missing some very fundamental elements required for understanding of the subject (6:39).

Why Do People Learn? People do not learn, at least not as rapidly or as effectively, unless they are trying to achieve a goal. A person must be motivated to learn either externally or internally. For the purposes of this research the author studied what motivates adults to learn since all AFIT students are adults.

According to Ron and Susan Zemke in their article "30 Things We Know for Sure About Adult Learning," there are six main reasons adults pursue education.

1. Adults pursue a new learning experience when they experience a specific change in their life such as divorce, a new job, a promotion, or the death of a family member. Education is a conduit to help the person cope with this change.

2. The more changes in an adult's life, the greater the chances are that a person will pursue education of some type.

3. Learning experiences that an adult looks for usually are related to the life change the person is experiencing (i.e. training for a new job).

4. Adults generally get involved in these learning experiences before, during, and after a significant change in their life. Once the change is assured, they tend to engage in any learning experience that appears to assist their coping mechanism.

5. *Adults pursue education as a means* to an end; they have a use for the education pursued.

6. Learning a new skill or enhancing an old one builds up the adult's self-esteem and gives pleasure to the adult. (26:58)

How Do People Learn? Learning is known to be a personal ability. No one can force another to learn. Teachers can not force students to learn. The student is the only person who can manage his/her own learning experience. Even so, researchers have learned much about what type of instruction is most effective.

David Ausubel believes learning can be segmented into two categories: **rote or meaningful**. **Rote** learning is pure memorization. This type of learning seldom is incorporated effectively into one's inner knowledge structure. On the other hand, **meaningful** learning builds upon knowledge already within one's inner knowledge

structure. Meaningful learning relates new ideas and concepts, with ideas and concepts already familiar to the person. (16:7; 11:2; 4:12) Meaningful learning is lasting learning. This is the type of learning one must strive for if the objective is to retain what one learns. Is it any wonder that learning that has a direct application in the workplace is the most successful long term retention learning experience (meaningful learning) (12:36)? Work related learning builds upon the employee's current knowledge level. People acquire new knowledge by starting with the simple and building up to the more complex (6:39). If one starts with the complex without first learning the simple, the learning experience will likely end in frustration because the framework for this knowledge is incomplete.

Is learning always initially successful? The simple answer is no. In fact, learning requires the freedom to fail. A person must recognize that he/she may fail at first. That is why it is important to create a situation where initial failure is acceptable in the training environment. Successive, small goals are best instead of unreasonably large goals which may discourage the individual and cause him/her to lose interest in the learning experience. Failure will lead to success in the end (12:36).

Teaching for Retention.

What is Teaching? Before one can discuss how to teach for retention, one must first understand the definition of teaching. This author has borrowed this definition from David Ryans article Evaluation and the Improvement of Instruction. "Teaching is the task of selecting, organizing, and forwarding information to the receiver" (19:3). In this context, the receiver is the student.

Environment. The surroundings and atmosphere influence learning potential. Ron and Susan Zemke offer the following guidelines for adult education based on their research:

1) "The learning environment must be physically and psychologically comfortable." Long class periods and long periods of sitting as well as scant practice time irritate adults.

2) Adults tend to be self-conscious and afraid of losing their self-esteem when asked to accomplish a task in front of the class.

3) It is important to address the adult learner's and teacher's expectations early in the learning process so that they are not at odds.

4) Adults have great experiences which can be tapped to benefit the learning opportunity.

5) New knowledge must be integrated with old knowledge (26:60).

A good learning environment allows for failure. Failure is normal at first and should be expected. Only with failure, can one succeed. (12:36)

Curriculum. What is curriculum? For the purposes of this research, the author defines curriculum as the information to be transferred from the sender (teacher) to the receiver (student) in a learning situation to help the receiver understand new ideas and concepts. How is the curriculum developed? What is known about curriculum design for adults?

Ron and Susan Zemke in, "30 Things We Know For Sure About Adult Learning" say there are several things researchers definitely know about building curriculum for adult education. Adult learners don't like survey courses. They instead tend to like single-theory courses that focus on relevant applications. Adults need to

be able to integrate/relate new knowledge with old knowledge or else they probably will not really learn the new material. New knowledge that is at odds with old knowledge that the learner holds true and which forces the learner to re-evaluate that old knowledge is harder to integrate into permanent memory. New information that has little conceptual overlap with what is already known is learned more slowly. Tasks that are fast-paced and complex probably will interfere with the concepts they intend to illustrate. Even though adults tend to acquire new information more slowly than children, adults tend to be more accurate and therefore rely little on trial-and-error. Adults are more self-conscious and thus take errors more seriously because they may damage one's self-esteem. Adults prefer self-paced projects. This does not mean they prefer to learn solely on their own. Instead adults tend to be more receptive to multi-media learning. Adults are efficiency minded; they want the cheapest, easiest, fastest way to learn the concepts (26:58-60).

How Is Effective Curriculum Built? Experts agree that curriculum is built using the familiar "Problem Solving" theory. The first step is to define the problem. In the learning context, the problem is how to get the information from the sender (teacher) to the receiver (student). Once the problem is defined, one must analyze the problem. One might ask is this issue really a problem? Once finished analyzing the problem one must then research the problem. Research includes investigating the body of knowledge available on the issue. This might be accomplished through a literature search. Next, one must determine alternatives approaches to solving the problem. What media should one use to get the new concepts across? How should the concepts be presented? When? Where? Once the alternatives are established, the instructor must test the alternatives. This may be

done by studying again the literature to determine if research has already been accomplished on the topic and determine the results of the research. If research has not been done on a particular alternative one might desire to construct an experiment to test the alternative. Once the alternatives have been sufficiently tested the instructor can now intelligently pick the best (or maybe most feasible) solution and implement the appropriate curriculum. The last and probably most important phase is to evaluate the results. Listen, listen, listen. Feedback is invaluable if one wants to determine success of failure of the curriculum. This feedback may force the instructor back to the beginning of the problem solving method once again if the problem is not solved. Changes may be needed to solve the problem. Changes are good if the change benefits the curriculum. One must remember that learning and teaching are iterative and evolutionary processes. (4) Now that the author has built a basis for discussion, the next step is to look at computer education.

Computer Education

Who needs it? The answer is "everyone". Much emphasis is given to the type of computer to buy, the accessories (peripherals), the software, and even the furniture for a new computer. Strangely enough, training is rarely a top priority. So how does a person learn to use the computer? Major James McConeghy, the USAF's small computer program manager for the Air Force Small Computer/Office Automation Service Organization (AFSCOASO) said when the small computers started arriving from the vendor he did not plan for training. His office understood from the various vendors that the computers were simple machines and so people didn't need training. The problem with that statement is that computers and software are becoming more

sophisticated (22:63-65). Simply providing employees with the new equipment and programs is not enough. Brad Bass wrote in a column in Government Computer News, "The majority of persons surveyed reported four major problems facing micro users. The most widely reported was the lack of adequate training." (2:61) A survey done by Ryan Nelson and Paul Cheney revealed that many companies are not meeting the computer training needs of their people. Technology gets priority, not training (15:121). Many people can remember seeing some of the Apple Computer commercials with the frustrated DOS user who takes a chain saw or sledgehammer to his computer due to frustration because he couldn't get the machine to do some simple work. Even though this ad was obviously a marketing exaggeration, the frustration described in the ad does exist if the user does not have proper training to do his/her job (13:52). In a This frustration leads to anxiety related to computers and lower productivity. One can hardly expect to have a high productivity level when one is afraid to use the means available to improve productivity. As such, computer anxiety is an important consideration when structuring a computer literacy course.

Computer Anxiety.

What is it? Computer phobia, Cyberphobia, Siliconitis, are all labels for a common disorder more simply described as "fear of computers." One would think that since computers have proliferated in most aspects of daily life that this phenomenon would be a thing of the past. Research indicates computer phobias are still common today. (5:527; 3:63; 8:16-17)

Symptoms of Computer Phobia. This disorder usually appears as a negative attitude toward the computer and could be any of the following:

- resistance to talking or thinking about computers

- fear of touching the computer
- feelings that one's intelligence is threatened by others who know something about computers
- expression of negative attitudes toward the computer
- feeling that the computer is easily broken and therefore one abstains from its use (3:64)

Causes of Computer Phobia. Computer phobia has many causes which can vary from person to person. Greg Brownell, in his book Computers and Teaching, remarks that this disorder is caused mainly by one of three problems:

- failure of the "phobic" to "keep up" with technology
- failure of the institution where the phobic works to keep the employee in the loop when making automation decisions and then pushing the technology onto the phobic
- failure of the institution to provide job incentives to individuals to keep up with technology (3:64)

Cures for Computer Phobia. Thomas Flygare, author of Cyberphobia Redux states that in a Lehigh University study of computer phobia, the only variable that appeared to have a significant relationship with initial computer anxiety was, previous experience with computers (5:527-8). Brownell believes that the cure for the phobia must come from the source. For example the phobic could enroll in a computer course to "get comfortable" with the new technology, or the institution could provide incentives for the employee to learn new technology, such as pay raises or both (3:64). The key to purging the fear is exposure and familiarity with technology. The more the phobic is exposed to the technology, the less anxious the person will be.

(3:64; 5:528) Training in the proper use of computers helps to reduce this anxiety. Surveys prove that users are more comfortable using computers once they are adequately trained (15:122).

What to Look For in Computer Training. User training must be targeted for specific needs and applications. Training for the data processing professional is inappropriate for the novice; likewise, advanced applications training is inappropriate if the trainee has not completed introductory and intermediate training. The instructor must know his/her audience to effectively transfer data to the end-user. For the purposes of this research, end-user is defined as a person who operates the computer.

Jean Ussery and Bill Marriott, in their article "What To Look For In End-User Training," discuss the importance of problem oriented training. "The end-user's approach to the computer is usually problem-oriented. When an end-user needs or wants training, he is usually driven by a problem he is trying to solve." (24:20) As such, training aimed at the end-user should be presented in problem solving format. Good training must stimulate the student to learn (7:17). Problem oriented training does this. The student is able to relate the training to something useful in his life.

Checklist for Good Computer Training. Figure 1 shows Ussery and Marriott's checklist for computer learning. It is divided into areas that the informed computer training shopper should consider prior to selecting any computer course.

Audience Specification. A good computer course should have a clear statement of who the course is intended for. If a course does not have such a statement, one should think twice before attending. Courses that purport to be a panacea for everyone's computer training needs probably have no clear audience in mind and may not be suited for all computer aptitude levels (24:21).

Prerequisites. Course prerequisites define what the audience should know prior to attending the course. For example, this may include such things

	EXCELLENT	GOOD	FAIR	POOR	POINTS
AUDIENCE SPECIFICATION DOES THE AUDIENCE DESCRIPTION FIT YOUR TRAINEES?	10	7	4	0	
PREREQUISITIES DO THE TRAINING MATERIALS LIST SPECIFIC PREREQUISITE SKILLS THAT ARE MET BY YOUR END USERS?	10	7	4	0	
PERFORMANCE OBJECTIVES ARE THERE PERFORMANCE-ORIENTED OBJECTIVES THAT DESCRIBE NEW SKILLS YOUR TRAINEES NEED?	10	7	4	0	
ADVANCE ORGANIZERS ARE THERE ADVANCE ORGANIZERS FOR COURSE SEGMENTS TO SERVE AS ROADMAPS FOR THE LEARNER?	10	7	4	0	
SAMPLE PROBLEMS ARE THE SAMPLE PROBLEMS DRAWN FROM TYPICAL TRAINEE EXPERIENCES?	10	7	4	0	
USER LANGUAGE DO THE TRAINING MATERIALS CONTAIN EVERYDAY LANGUAGE AND DEFINITIONS?	10	7	4	0	
MODULAR DESIGN IS THE COURSE DESIGNED IN MODULES OR SEGMENTS PROVIDING FLEXIBILITY AND REVIEW?	10	7	4	0	
TOPIC SELECTION DOES THE DESIGN OF THE COURSE ALLOW THE TRAINEE TO SELECT SPECIFIC TOPICS?	10	7	4	0	
REVIEW DOES THE TRAINING INCORPORATE PLENTY OF REVIEW?	10	7	4	0	
PRACTICE IS THERE FREQUENT PRACTICE AND FEEDBACK FOR EACH NEW SKILL?	10	7	4	0	
				TOTAL POINT VALUE:	

Figure 1: Computer Training Checklist (24:21)

as ability to type, or maybe knowing the primary components of a computer. The prerequisites should be directly measurable rather than conceptual. Be wary of a prerequisite that says "understand" or "be familiar with." (24:21)

Performance-Oriented Objectives. Objectives should be measurable in terms of what the student will learn. Again be wary of "understand" type objectives which are hard to measure. Performance oriented objectives state

- who does the learning

- what specific tasks the student will learn

- under what conditions the learner will be able to accomplish each task

- what level of performance the student can achieve (24:21)

Advance Organizers. Simply stated, advance organizers are outlines at the beginning of each segment to act as a road map for the student showing where he will go during the training. This helps organize the student's learning experience and fills in the blanks during the course. (24:21)

Sample Problems From Real Life. One should learn by example. Training that relates to concepts already known promotes knowledge that is retained. (24:21)

User-oriented Language. The language of data processing is rarely understood by the novice. The trainee must learn some of the language, not all of it. Good training will support this premise, not hinder the learning of the student. (24:21)

Modular Design. Learning a new skill in small packets instead of all at once helps the student learn. Small, manageable units of related material presented and practiced over time helps the student retain more. (24:22)

Review. Review of concepts helps the student build lasting knowledge. It also gives the student the opportunity to examine what has been learned and assures understanding the concepts.

Practice. New skills practiced over time last longer. If a student does not have the opportunity to practice new skills he/she probably will not be able to accomplish tasks on his/her own. Practice gives the student a chance to show

the instructor what he/she knows. Feedback from the instructor reinforces the skills. (24:22)

Methods of Computer Training. There is a myriad of possibilities. One could learn on one's own using the software manuals, use Computer Aided Instruction, Videodisc workstations, peer tutoring, attend formal lecture type classes with or without hands-on opportunities, or a combination of any of the above. Each of the above has advantages and disadvantages. These are discussed below.

Manuals. Learning from manuals is usually the worst method of learning computer applications. Manuals are difficult to understand and when things go wrong there's nobody there to help the trainee (8:20; 18:147). In John McPartlin's article "The Trouble With PC Training," he compares reading and learning from software manuals to studying Sanskrit. They fall into the category "kids don't try this" arena (13:52). Many times, this type of training causes anxiety from the experience and is counterproductive to reducing cyberphobia. Training is meant to reduce the initial hurdle in computing and give the trainee more confidence in using the computer (18:147). The only real benefit to this type of training is cost. Once the trainee has the software and the manuals, the only investment is time.

Computer Aided Instruction (CAI). The title may be a bit intimidating. Computer Aided Instruction is just a fancy word for "computer-based tutorial." In this context a computer-based tutorial is a series of computer programs used to present information to aid learning of computer related material. The benefit of this type of training is the user has ultimate control over when, where, how long, and what to study at any given time. The user is not obligated to attend a class. This type of instruction media is particularly beneficial when other methods of instruction are not

available or the user's schedule does not permit attending a structured class. The problem with this method is that most tutorials are designed for the beginner. Research has shown that this category or user can better benefit from the structured classroom environment (discussed later). In addition, CAI tends to be generic in its approach so it can't be modified to suit a particular situation. This is not to say that CAI is not a valuable tool for learning computer material; it is only to say that is in not the most effective means.

Videodisc Workstations. This approach is gaining popularity in the training arena. Videodisc instruction couples a disk-based lesson with a manual and a video tape. The user watches the video to learn how to run the application. During the video, the trainee is encouraged to turn off the tape between lessons and practice what has been presented. The benefit of this type of media is the trainee has hands-on time to practice the activity and can review the tape as often as desired until the lesson is understood. The drawback is this type of media is expensive and the student still does not have anyone to answer questions. This method is most effective for introductory training where the concepts are basic. As complexity grows, this method does not lend itself to effective instruction (18:152). Even so, this is a good alternative as the video can be used as a reinforcer for the trainee. (10:113)

In general, the extent to which self-paced training can be helpful varies according to the motivation level of the trainee. If the student is highly motivated or has an immediate need for the computer knowledge, then self paced learning can be effective.

Peer Tutoring. This method involves learning from a colleague. Although this method is cheap, it is not recommended for initial training. The major

problem with this type of training is that the trainee is not likely to learn the full capability of the application. If the colleague that is teaching learned this way, that person is likely to only be familiar with things he uses frequently.

Classroom Training. The advantages of classroom training are many. Most people who are new to the computer prefer a live person demonstrating the functions, describing difficult concepts, and answering questions. Instructors can bring the training experience to life. The instructor can "read" his/her audience and relate to the individuals in the class. Concepts are easier to understand when the student can stop the instructor and ask questions. In addition, the instructor can tailor the lesson to the needs of the class. This advantage is especially important as the concepts get more complex. For these reasons, classroom instruction is preferred. The major disadvantages of this type of training are cost and time spent away from the trainee's duties. (18:148)

The question of hands-on instruction versus lecture is an important one. Typically, hands-on instruction is most effective because it gives the student a chance to try-out what the instructor is teaching. The student can practice his skills regularly in the presence of the instructor. Richard Riddington, in his article "Selective Shopping Yields Greatest Training Payback," likens hands-off training to a magazine review, "You still don't get skills, and people don't want to pay high course fees for what they can get from a magazine." (14:57)

Researchers say hands-on classroom instruction and practicing helps the student retain what is taught (17:22; 18:148).

Lecture type training is usually used when the audience is large and there are not enough resources for each student to have a computer to practice. This type of

training is less effective than hands-on because the student cannot get the immediate feedback he gets from a hands-on session. It also typically bores people, and as a result, people find it hard to learn the concepts. (18:148)

The bottom line is that there are many alternatives to learning computer applications. The teacher must evaluate the alternatives and make decisions accordingly.

Avoiding Overload.

How Long. How long is long enough for a training period? Most researchers agree that too much computer training in a single session causes computer overload. As with any learning experience, the student must be able to transfer what is held in short-term memory to permanent memory for it to last. If the training session is inordinately long, the student becomes overwhelmed and the learning experience fails to achieve its goals.

Computer training is extremely susceptible to this problem because most computer instruction requires repeated practice in order to master a concept (17:22). For this reason, researchers believe rather than trying to teach a computer application in a single eight-hour session, it is better to split the class into eight one-hour sessions. This helps eliminate computer overload and gives the student the time needed to really learn and retain the information (17:22; 10:112; 7:17). Computer learning takes time.

How Much. John Hansgate in his article "Avoiding Overload," relates a personal computer training experience his company had. The training plan they used was too ambitious. They failed to read their audience and determine their needs. As a result, after the class, their support staff continued to receive daily calls to explain functions and operations that were taught in the class just days before. The

problem was the class tried to cram too much in too short a time. They decided to do something about it, so they reevaluated the curriculum and restructured the class into introductory, intermediate, and advanced level material. The introductory material included the basics to get the students started. It dealt with the least complex concepts and was taught in such a way that it formed the groundwork for each of the other courses. The intermediate course introduced some of the power of the applications. This course built on the concepts taught in the introductory course. The instructors used work-related exercises to stimulate the students. Finally, the advanced course introduced the real power of the software. The next time the class was taught, the students retained more of the material and the telephone calls to the support staff diminished. Figure 2 shows the instruction outline used.

Considering the time and effort invested in using software to do one's job, it seems only logical that the instructor must structure the course for maximum learning retention. As Hansgate says, "Good training must stimulate curiosity, allay apprehension and, above all, avoid presenting an overwhelming task." (7:18)

Summary

This literature review explored the meaning and importance of computer literacy as it applies to the world today. It is obvious that computer literacy requires computer training. The author presented information about basic learning theory to build a basis for discussion about computer education today. The author then discussed today's computer education and how best to transfer knowledge from the instructor to the student.

This literature sets the background for the author to research the type of training necessary for AFIT students and to design a computer literacy course for these same students.

	INTRODUCTORY	INTERMEDIATE	ADVANCED
OPERATING SYSTEMS	PURPOSE KEYBOARD LAYOUT HARDWARE INTERFACE SYSTEM BOOTING CRITICAL FILES BASIC COMMANDS ON-LINE/ON-FILE FILE MANAGEMENT DIRECTORY MANAGEMENT LOOK AHEAD	EDITOR USE BATCH FILES SYSTEM INTERFACE COMMANDS SELF-BOOTING APPLICATIONS HARD DISK MANAGEMENT	ADVANCED EDITOR ADVANCED BATCH FILES
WORD PROCESSORS	PURPOSE KEYBOARD LAYOUT CURSOR CONTROL COMMAND MENU TEXT FORMATTING TEXT ENHANCEMENTS TEXT EDITING SPECIAL COMMANDS PRINTING LOOK AHEAD	ADVANCED FORMATTING ADVANCED EDITING FILE MERGE MAIL MERGE FORMAT FILES PERSONAL DICTIONARY	APPLICATIONS DEVELOPMENT MACROS
SPREADSHEETS	PURPOSE COLUMNS & ROWS CELL ADDRESSING RELATIVE AND ABSOLUTE COMMAND MENU CELL ENTRIES BASIC FUNCTIONS STATISTICAL & ARITHMETIC SAMPLE LAYOUTS WHAT-IF TRIALS LOOK AHEAD	WINDOWS FILE LINKING SPECIAL FUNCTIONS BOOLEAN LOGIC & STRING FUNCTIONS INTEGRATED FUNCTIONS DATABASE AND GRAPHICS BASIC MACROS WORK SAVERS	APPLICATIONS DEVELOPMENT FILE CONVERSION ADVANCED MACROS PROGRAMMING & SELF-EXECUTING
DATABASES	PURPOSE FILES, RECORDS AND FIELDS FILE DREATION FILE MAINTENANCE AD HOC COMMANDS DATA EXTRACTION SYSTEM CONTROL REPORT GENERATION LOOK AHEAD	ADVANCED DATA EXTRACTION CONDITIONAL & QUANTITATIVE INTRODUCTION TO APPLICATIONS PROGRAMMING & VARIABLES ADVANCED REPROT GENERATION MULTIPLE FILES	APPLICATIONS DEVELOPMENT DATA COMPUTATIONS CONTROL BREAKS BRANCHING DO-LOOPS FORMATTING

Figure 2: Microcomputer Software Instruction Outline (6:18)

III. Implementation/Analysis

Introduction

The author used a combination of personal interviews, and a review of applicable computer literacy/training literature to develop the curricula. In addition, the author taught one section of QMGT 290 and evaluated the end-of-course critiques.

As a prerequisite to all research, the author operationally defined computer literacy in the academic environment using personal interviews and a literature review of professional literature dealing with computer literacy and basic learning theory. The researcher used the literature review to set a stage for computer education by first discussing general concepts of learning theory and then specifically discussing current data pertaining to computer education.

The author answered question 1 through discussions with the AFIT Graduate program degree area Program Managers. These faculty members are responsible for the different courses that students must take at AFIT to fulfill the requirements for a masters degree in that area. Question 2 was answered through interviews discussions with the instructors of each required course that uses computers to accomplish course objectives. Question 3 was answered through a combination of literature review; discussions with Lieutenant Colonel Richard Peschke, QMGT 290, Introduction to AFIT Computers, course director; Lieutenant Donald McNeeley (USN) and Major Jake Simons, QMGT 290 course instructors; and Daniel Reynolds, AFIT Mathematics department instructor; and discussions with Ms Charlotte Fordyce, Wright State University CS205, Computer Literacy for Business Students. The Author set the

baseline with the literature review. With the baseline defined, the author discussed the requirements with Ms Fordyce and made recommendations to the AFIT faculty. The author had further discussions with Lieutenant Colonel Richard Peschke on course content and then proposed a draft syllabus to the instructors for QMGT 290. Question 4 was answered through a combination of literature review of the applicable research in this area and discussions with Lieutenant Colonel Richard Peschke, and Ms Charlotte Fordyce. The discussions with Ms Fordyce were used to determine the extent and approach to computer training at Wright State University.

This information was used to propose a course syllabus for a revised "Introduction to AFIT Computers" course that was introduced during the 1990 Summer short term at AFIT. The author presented the revised course to one section of QMGT 290 in order to get first hand feedback from the students.

The success or failure of the revised course was examined by reviewing the end-of-course critiques completed by AFIT Class 91S/D. The author evaluated the entire data set of critiques.

Course Planning

The author worked with the QMGT 290 course director and instructors to develop the course plan. The basic shell was derived from John Hansgate's article Avoiding Overload (7:17). Figure 3 shows this shell in detail. The figure lists, by topic area, the structure of a course covering Operating System, Word Processing, Spreadsheets, and Data Bases at three aptitude levels. Since QMGT 290 does not cover the Operating System and Data Bases, these areas were not included in the proposed course plan. This is not to say that learning the Operating System and Data

	INTRODUCTORY	INTERMEDIATE	ADVANCED
OPERATING SYSTEMS	PURPOSE KEYBOARD LAYOUT HARDWARE INTERFACE SYSTEM BOOTING CRITICAL FILES BASIC COMMANDS ON-LINE/ON-FILE FILE MANAGEMENT DIRECTORY MANAGEMENT LOOK AHEAD	EDITOR USE BATCH FILES SYSTEM INTERFACE COMMANDS SELF-BOOTING APPLICATIONS HARD DISK MANAGEMENT	ADVANCED EDITOR ADVANCED BATCH FILES
WORD PROCESSORS	PURPOSE KEYBOARD LAYOUT CURSOR CONTROL COMMAND MENU TEXT FORMATTING TEXT ENHANCEMENTS TEXT EDITING SPECIAL COMMANDS PRINTING LOOK AHEAD	ADVANCED FORMATTING ADVANCED EDITING FILE MERGE MAIL MERGE FORMAT FILES PERSONAL DICTIONARY	APPLICATIONS DEVELOPMENT MACROS
SPREADSHEETS	PURPOSE COLUMNS & ROWS CELL ADDRESSING RELATIVE AND ABSOLUTE COMMAND MENU CELL ENTRIES BASIC FUNCTIONS STATISTICAL & ARITHMETIC SAMPLE LAYOUTS WHAT-IF TRIALS LOOK AHEAD	WINDOWS FILE LINKING SPECIAL FUNCTIONS BOOLEAN LOGIC & STRING FUNCTIONS INTEGRATED FUNCTIONS DATABASE AND GRAPHICS BASIC MACROS WORK SAVERS	APPLICATIONS DEVELOPMENT FILE CONVERSION ADVANCED MACROS PROGRAMMING & SELF-EXECUTING
DATABASES	PURPOSE FILES, RECORDS AND FIELDS FILE DREATION FILE MAINTENANCE AD HOC COMMANDS DATA EXTRACTION SYSTEM CONTROL REPORT GENERATION LOOK AHEAD	ADVANCED DATA EXTRACTION CONDITIONAL & QUANTITATIVE INTRODUCTION TO APPLICATIONS PROGRAMMING & VARIABLES ADVANCED REPROT GENERATION MULTIPLE FILES	APPLICATIONS DEVELOPMENT DATA COMPUTATIONS CONTROL BREAKS BRANCHING DO-LOOPS FORMATTING

Figure 3: Microcomputer Software Instruction Outline (6:18)

Bases are not important areas of computer training. The consensus of the course director and the instructors was that, since there was only four weeks to teach this course, the course would focus on the applications software used at AFIT.

The course director felt that learning the operating system would be the responsibility of each student and as such the course would assume minimum operating system competency. Minimum operating system competency is defined as the ability to get a microcomputer started and running, format a blank diskette, copy files, delete files, use floppy disk drives for file management, and use hard disk drives for file management. To aid the students in attaining minimum competency, an operating system tutorial (MRDOS) was distributed to all new students with the newcomer packages in the Spring prior to arrival at AFIT. MRDOS is a self paced, computer based, MS-DOS operating system tutorial available as shareware from major Bulletin Boards such as EXEC-PC and SIMTEL-20. MS-DOS is the standard operating system for IBM-PC computers and compatibles. This operating system is installed on all AFIT microcomputers. Each MRDOS diskette was distributed with an introductory letter (see appendix A) telling the student about MRDOS and giving basic operating instructions for use of the software. The tutorial included basic and advanced operating system instruction with end-of-lesson quizzes to test the user's knowledge of MS-DOS. Students were advised that operating systems would not be discussed as a formal part of QMGT 290 and advised to use MR-DOS to become familiar with MS-DOS.

QMGT290 Course Objectives. QMGT 290 was structured to provide new students with the a beginning knowledge of each of these packages. The objective of the course as documented in the course syllabus was to provide a basic understanding

of Mathcad™, Quattro Pro™ and WordPerfect™ so that the students can begin to use these packages in the graduate program. In addition, a collateral objective was to understand how these three software packages could be used in concert by sharing data between the packages to assist managers make decisions. Based on the objectives, the course content was developed to provide introductory level skills.

QMGT290 Course Syllabus. The QMGT 290 syllabus was developed around three applications software packages; Mathcad™, Quattro Pro™, and WordPerfect™. Mathcad™ is a product of Mathsoft Corporation. It is a quantitatively oriented software package designed for mathematical analysis, used at AFIT in MATH290 and MATH314; Math Review; EMGT290, Math/Computer Review for Engineering Managers, courses during the Summer short term and in MA591, Statistics I, and MA691, Statistics II graduate courses at AFIT. Quattro Pro™ is a spreadsheet product of Borland International which is compatible with Lotus 1-2-3™ and is required for LOGM 510, Introduction to Spreadsheets and Database Management. It is highly recommended in MA592/692, Statistics and is used in other courses such as LOGM590, Computer Applications for Managers and OPER 526, Quantitative Decision Making. WordPerfect™ is a word processor from WordPerfect Corporation. It is the choice of many graduate students for preparing written course assignments and thesis documentation. WordPerfect™ is the word processor available to students who desire to use institute equipment for accomplishing written assignments.

Since the literature did not address mathematical software programs, the author discussed course content with Daniel Reynolds, Assistant Professor of Mathematics and Statistics and instructor for MATH314, Math Review, to determine those skills necessary to begin using Mathcad™. Professor Reynolds' course is more

intensive than MATH290. The author, and the course director for QMGT290, felt that Professor Reynolds was the most appropriate instructor to set the baseline for the Mathcad™ instruction since his classes use the program most intensely. Once the baseline was set, the other MATH290 instructors were consulted to assure that the baseline was appropriate to prepare their sections of the MATH290 course with the necessary Mathcad™ knowledge. It was determined that the basic Mathcad™ skills include the following:

Hardware Installation

Printer Selection

Basic Entry

Basic Functions

File Management Commands

Basic Output

Professor Reynolds felt that these skills would prepare the student to begin using Mathcad™ in MATH314 during the summer short term. As mentioned above, all other segments of QMGT290 were structured after the Hansgate introductory instruction model. The revised QMGT290 course syllabus is at Appendix B. Course charts for each module of the course are shown at Appendix C.

Course Delivery

Upon completion of the course planning, three AFIT instructors, Lieutenant Colonel Peschke, Lieutenant Donald McNeeley (USN), and Major Jake Simons; as well as the author, presented the course material to the new AFIT graduate students.

Teaching Methods. The instructors had many options available to them for presentation including hands-on, video instruction, lecture with a live instructor computer aided instruction (CAI), peer tutoring, manuals, and any number of combinations of these approaches. The instructors elected to use a combination of class lecture, hands-on in-class practice and out-of-class hand-in group projects to present the material and reinforce the concepts. The decision was based on research cited in Chapter 2, stating the most effective method of computer training is hands-on with an instructor.

Lecture. Each class commenced with a lecture to orient the students to the topic of the day's discussion. This was facilitated with the use of prepared charts (overheads) and through overhead projection of the instructor's computer session via a simple Black and White projection display panel. The overhead projection of the instructor's computer session allowed the students to watch the instructor actually perform the commands in real-time. In this way the students were able to observe the desired behavior before performing it themselves.

Hands-On. Students were then encouraged to practice the concepts with their own computer. They were assigned two to a computer system. One student per computer system was not practical, given the number of systems available, the total number of hours available in the computer labs, and the number of students requiring instruction. As such the students were encouraged to share the hands-on time to assure all students had the opportunity to perform the desired tasks.

Out-of-Class Projects. Out-of-class, hand-in group projects were used as a method of review to solidify the concepts demonstrated and practiced in class. Students were asked to divide themselves into two person groups to accomplish the

projects. The group approach was used to promote student interaction and stimulate thought about the solution approach. Groups take advantage of the synergy possible when more than one person works on a project. In addition, the group concept promotes peer tutoring which tends to reinforce the instruction and promote a better understanding of the subject.

This methodology is in concert with the research that concludes that classroom hands-on instruction with a live instructor tends to be the most effective means to teach computer skills. The in-class examples and out-of-class practice projects were structured around samples from real-life to enable the students to draw from their experiences thereby linking the learning to real life. This, again, is a basic concept of learning theory. It is proven that a person tends to retain more when new knowledge is associated with old knowledge. The projects used included a loan repayment problem not unlike one that most students have probably experienced when buying a car; a class grade evaluation problem; a learning curve problem commonly associated with a production organization; and a management operations problem, also associated with a production organization. Each problem was tailored to resemble a real-life scenario. Examples of the class exercises and projects are in Appendix D along with sample solutions. All projects and problems conformed to the guidelines set forth by Jean Ussery and Bill Marriott, in their article "What To Look For In End-User Training" (see Chapter 2 for further discussion).

Class Logistics

Section Composition. The author, course director and course instructors evaluated three different methods of class composition: grouping by major program

(GSM, GLM, GEM, GCM, GCA, GIR); grouping by option (GSM, GTR, GIM, GEM,GCA,GCM,GIR); grouping by aptitude; and random grouping.

Grouping by Major Program. Grouping by major program had the advantage of reducing the number of sections required as compared to grouping by option. In addition, this method would allow the course to be tailored more specifically to the audience wherein the exercises and projects could be tailored toward the background of the section. This approach also offered the greatest ease for the schedulers. On the other hand, the students do not get the benefit of interaction with students from other career fields. This interaction can introduce additional creative insights for the software taught.

Grouping by Option. This strategy would provide the best environment to tailor a course toward a particular audience since the option breakout is the lowest level identification unit for the school. For example, the GLM program includes the following options: Transportation Managers, the Supply Managers, and the Item Managers. Grouping by option would allow very specific instruction toward the audience by introducing real-life problems from the specific area of expertise. As an example, the Item Managers may be given an exercise that demonstrates the use of the spreadsheet for solving an Economic Order Quantity problem. The transportation managers may be given an exercise demonstrating how to solve a distribution problem. The item managers may be given an exercise demonstrating how to solve a database type problem.

The major disadvantages of this approach include unevenly sized sections (more sections) since some options only have a few persons, and reduced variety in the classroom since the audience would have a common background.

Grouping by Aptitude. Grouping by aptitude allows the instructor to differentiate the instruction based on the section aptitude. This could lessen the boredom felt by some of the more experienced students while the instructor is focusing on bringing the rest of the class up to a minimum competency level. This would require up to three different courses each focusing at a different level of student computer aptitude. In addition, the course director would have to devise a method of pretesting the students to determine which level was appropriate for the student. Since the course includes instruction on Mathcad™, Quattro Pro™, and WordPerfect™, the faculty would be required to devise separate tests for each package. Grouping by aptitude would further require either several additional sections or grouping sections strictly by aptitude without regard for the major or option since there could be as many as 84 different combinations (3 different levels for each of three courses, in which a student could be in any one of the three levels in any one of three classes) of sections. The principal disadvantage though is that by segregating the classes into aptitude levels, if there is not sufficient time in the program to bring all students to the same final level of competency, then the school may increase the disparity in overall aptitude levels instead of leveling them as desired.

Grouping Mixed. Mixed composition of classes has several advantages. This method promotes variety in the classroom which may introduce variety into the student's thoughts about how to approach a problem and also different uses for the software in the real world. In addition, class size could be leveled so that each section would have an equal number of students. This could avoid the overcrowding caused by some of the other grouping methods. Scheduling is simplified since any student would be a candidate for any class. Unfortunately, this method

would cause the class to be more generic since tailoring to a specific area of expertise would be impractical with the mixed group. This grouping strategy would not allow the instructor to focus on individual program needs.

As a result of the above analysis the classes were principally broken up according to graduate program (GSM, GLM, GIR, GCA, GCM). This approach was taken for two major reasons: subject matter and ease of scheduling. Since each graduate major program requires a different set of courses to be successfully completed in order to graduate, it made sense to break the sections in this manner thereby permitting the instructor to tailor the sessions toward the specific needs of the group. For instance, the two GSM sections were given more intensive Mathcad™ instruction since that curriculum requires greater use of that software. Some of the other majors use spreadsheets more. In addition, grouping by major made the scheduling task easier since all students in a particular major take the same courses at the same time (unless class size demands additional sections for a particular major). This approach required eight sections of QMGT290 in the summer 90 short term.

Scheduling.

Class Duration. The session duration was determined based on subject matter to be covered, availability of computer assets, availability of instructors, and duration of the summer short term. The limiting factors were the duration of the summer short term and the availability of computer assets. The course director was restricted to the four week time frame covered by the summer short term. Given this limitation the next consideration was the amount of time available in the labs. QMGT290 principally used the Graduate computer classroom (Room 315) for instruction. A second computer classroom was occupied during part of the short term

by courses from the Professional Continuing Education program. This made use of this classroom impractical on a continuing basis. In a normal work day there were eight hours of time available in the lab in Room 315. In addition, the Math Review course required access to the lab one day a week. This left four days a week and eight hours per day. The author, course director and course instructors determined that in order to minimize overloading the students in any one session, class sessions should be limited to two hours. This figure was arrived at by experience of the course director and the course instructors. Given a two hour timeframe per session, the sections were scheduled to meet twice a week for two hours each session. This arrangement made 100% use of the computer classroom throughout the week during the entire short term.

Teaching Approach. The author evaluated at two different options; using one instructor to teach only one application; and using one instructor to teach all applications.

One Instructor/One Application. The major advantage to this approach is continuity of instruction across all sections. With a single instructor teaching all sections of one application, that instructor can plan the class session so that all sections receive the same instruction (within reason). The instructor's lesson plans would be simplified since the instructor would prepare for only one application and focus efforts there. In addition, the instructor would therein specialize in one particular application and become the expert on that application instead of trying to be an expert in all three application programs. Unfortunately, there are disadvantages. First, the instructor may tend to become bored teaching the same subject over and over for eight

hours per day. In addition it requires the instructor to be well versed in that software since the entire new class would be depending on a single instructor.

One Instructor/Many Applications. This approach staves off the likelihood that the instructor will get bored with the topic being taught. In addition the instructor workload is spread over the four weeks instead of being concentrated in one week. Also a single instructor lends continuity of teaching style and may avoid the notion that the course is fragmented. On the other side, each instructor must be knowledgeable in all applications and there is no assurance of course continuity across the sections.

Considering the alternatives given, the author, course director, and course instructors opted for the one instructor/many applications approach. The decision was based both on instructor workload and continuity of teaching style. Each instructor was assigned two sections and the author taught one section with the remaining section taught by the GEM program manager.

Course Feedback

This section discusses both instructor and student feedback on the course. The author interviewed each QMGT290 course instructor to garner feedback relating to the success or failure of the course as presented. Feedback is based on that instructor's individual experience in the classroom and may not necessarily be indicative of the entire course. The author collected all student end-of-course critiques for QMGT 290 to determine student attitudes toward the revised course.

Instructor Feedback. The author interviewed all the QMGT290 instructors to gather instructor feedback on the revised QMGT290 course. The author has grouped

the instructor comments into broad categories for discussion. These categories include student general computer knowledge prior to class, class demographics, accomplishment of course objectives, and problem areas.

Student Knowledge. Each instructor was asked to describe the overall proficiency level of their sections prior to QMGT290. All four instructor agreed that the proficiency levels appear to be increasing overall in contrast with previous classes. More students appear to have had previous computer experience prior to arrival at AFIT than in the past. In fact all four instructors reported that over 75% of each section had some experience with computers. That is not to say that there are no computer novices entering AFIT; but, the number of novices is shrinking from previous years. Three of the four QMGT290 instructors reported greater than 50% of their sections were familiar with some type of spreadsheet and/or word processor. It appears that as computers continue to proliferate the workplace, the employees are getting more computer literate. Two of the four QMGT290 instructors reported that over 50% of the new students in their sections owned a personal computer prior to taking QMGT290.

Section Demographics. Each instructor was asked to describe the demographics of his particular sections. All instructors reported varied levels of proficiency within each section. Proficiency within each section ranged from computer novices to computer experts. One of the instructors characterized the proficiency levels as follows:

- Novices possess little or no previous computer knowledge. In many cases they are weak in basic DOS knowledge.

- Intermediate (medium) level proficiency seemed to characterize those students that know the basics of computers but lack applications software knowledge.
- The expert tends to be familiar with the overall workings of the computer as well as with applications software.

Due to the nature of scheduling of sections, no attempt was made to segregate sections into proficiency levels. Three of the four instructors felt the mixed nature of the sections introduced some discontent among the students. For instance, since the course assumed basic computer proficiency, the novices tended to get lost and at times expressed frustration with the course. The experts on the other hand, felt they gained little from the course. Three of the four instructors feel that the course director needs to develop some method of either testing-out students with sufficient knowledge in the applications area or structuring different levels of instruction for each module of the class. The course director does not advocate a test-out process. He feels that every student can benefit from the class. Instead he advocates developing modules of instruction aimed at different proficiency levels. Three of the four instructors feel three different levels of instruction would be adequate. The basic course would include such things as basic computer orientation (hardware and software), DOS, and a beginning level of applications software. All the instructors agreed the revised QMGT290 provided this year would be adequate for the intermediate level course. The expert level course would build upon the intermediate level course with more advanced instruction focusing on the power points of the three applications software packages introduced in the course. This might include advanced file management techniques, macro development and use, etc.

This instruction technique would not only provide focused instruction to each level of student; it would also likely reduce section sizes and increase the number of sections. The smaller class size could reduce the congestion in the computer classroom and possibly allow each student access to his/her own computer during class. As mentioned earlier, one student per computer is the optimal mix for effective computer learning.

The increase in the number of sections presents a potential problem with logistics. More sections means longer class days. At this offering of QMGT290 the graduate computer classroom was used continuously from 0800-1700 daily. Due to the constraint on the number of computers available to teach the course, it appears that increasing the class day would be the likely alternative. The course director commented that in previous years the class day had been extended into the evening hours.

One QMGT290 instructor does not feel that segregating students into proficiency levels is advisable. He contends that the objective of the course is to develop a class wide minimum proficiency level and if we segregate classes into different proficiency levels, AFIT will widen the computer proficiency gap among students. He contents this may be counterproductive in the long run. The other three instructors do not share this opinion and believe there is no requirement to "level" the proficiency of the student body. Instead, the focus is to ensure at least a minimum proficiency level.

Course Objectives. The author asked each instructor for feedback regarding accomplishment of the course objectives. All four instructors felt that the course met its objective of presenting the material to the students. All new students

were exposed to the three software packages. The level at which the material was presented varied among sections. Two of the four instructors spent three days on Mathcad™, two days on Quattro Pro™, and two days on WordPerfect™. The other two instructors spent less time on Mathcad™ and WordPerfect™ and substituted material on VAX VMS™ and terminal emulation in the remaining classes. All four instructors agreed that the course content was optimized for the time allotted to the course. Three of the four instructors felt the number of in-class hours should be increased in future offerings. These same three instructors felt the current course attempted to present too much material for the time allotted to the class. This feeling was echoed by the students that took the revised QMGT290. As such, these instructors are skeptical about the depth of student learning resulting from the revised course. At best, the course exposed the students to applications software used during the graduate program at AFIT. These three instructors felt the in-class course time should be as much as doubled (four, 1½ hours sessions per week). In addition, section size should be limited to one student per computer if possible to maximize hands-on time. If this is not possible, then two students per computer should be the maximum student to computer ratio.

The lone remaining QMGT290 instructor felt that the course objectives were met and the in-class time was sufficient to provide minimum proficiency in the applications programs presented. He felt that if the course was expanded, it should be expanded to include the essential elements of the current LOGM510 course (required in all but three graduate programs at AFIT). Then LOGM510 could be deleted from the graduate program.

Problem Areas. Each instructor was requested to identify any problem areas with the revised QMGT290. Most of the problem areas centered around class logistics (equipment, class time, class room availability).

One of the biggest problem areas was equipment related. During the four week period that QMGT290 was taught one computer classroom computer system was inoperative for most of the time. A single inoperative system causes an overload in the classroom due to class size. With one system broken, two additional systems must service three students vice two. The instructors felt that this level of computer to student ratio is not conducive to effective learning. The institute should develop contingency plans for maintenance of the classroom equipment to minimize impact on the course. These plans might include priority maintenance for classroom equipment (one day turn-around) or swapping classroom equipment with other equipment within the institute to ensure all systems are operative at all times in the classrooms.

Another common problem during the course was the lack of required device drivers on the classroom computers. These drivers control such things as the printer interface, screen interface, and utility programs, all which were required for the course. When software is installed on the classroom equipment it should be installed with a complete complement of applicable device drivers and utility programs so that classes are not interrupted due to lack of proper drivers and utility programs.

All instructors agreed that the graduate program needs to have priority scheduling for the two computer classrooms during the summer short term. This year the graduate program was not able to use the PCE computer classroom due to PCE courses requiring the equipment at the same time. As a result, each section of QMGT290 had to be limited to three hours per week (two, 1½ hours sessions). At that

rate the graduate computer classroom was occupied from 0800 - 1700 Monday through Friday. If the graduate program was given priority scheduling for the PCE computer classroom the course could be expanded as desired.

All instructors agreed that a common point of confusion to the students was the lack of menu standardization on AFIT computer systems. For instance the menu used in the graduate lab is Hard Disk Manager (HDM™), the menu in the graduate classroom is a generic AFIT-produced menu; some other systems use GMENU™. Each menu system operates differently so students must be familiar with each different menu system to be able to use the computers available to them. To complicate things further, many of the computers with HDM™ as their menu system are not configured alike, placing applications software into different submenus which further complicates the students task. A single, common menu system should be installed so that the students can focus on learning the applications software instead of struggling with a menu system before the student can get to the application software.

Student Feedback. Due to the magnitude and variety of comments on these critiques, the author extracted the most common comments and will consider each in turn.

The most common area of concern about the course dealt with class length and size. A large majority of the students responding stated that the class was not long enough to really learn much about each of the software applications packages. The students generally felt that the overall course objective was attained. However, consensus was that specific course objectives were too broad for the time allotted to the class. The students appear to desire more in depth lecture and practice in each module with more Air Force oriented applications. Most students felt that the examples

used were appropriate but they wished there were more simple (30 minute) examples with an Air Force orientation.

Many students felt that the course should range between six hours weekly up to as high as 10 hours weekly. Class size generally was limited to provide one computer system for every two students. A large number of students felt the class size was too large to enable effective learning. The majority of students felt the class size was at the maximum level practical for teaching computer skills.

Another common comment was that in the classes where one was available, a lab assistant helped the instructor balance his time with each student. Many students expressed desire to include other applications in the course including VMS operating system, DOS, Graphics and Telecommunications. The students realize that in order to include these topics, the course time would have to be greatly increased.

Another common comment was that since the sections were mainly defined along graduate program lines (GSM, GLM, GIR, GEM, etc) the classes tended to have a wide variety of aptitudes. Many students felt that the more experienced students tended to be bored and got very little learning from the class. The students with little or no previous experience with computers, on the other hand, felt overwhelmed by the pace of the class. Many of the beginners felt they had to spend an inordinate amount of time on the out-of-class projects because they lacked a basic knowledge of computers. A number of students expressed desire for a test-out procedure for the course. They felt that by testing out students with adequate beginning level knowledge, the instructor would have more time to spend on the basics with the beginners. Still other students desired a more advanced study of each software package.

A common criticism across all sections was the desire for a reference text to accompany the class. Many students thought the class should have designated a reference text, available for purchase from the AFIT bookstore. Students felt they had problems they had difficulty solving because they did not have the software manufacture's manuals or a good second source manual for reference. Many students advocate having manuals available in all labs and classrooms for each software package. Another thought was to have the manuals and other reference sources on reserve in the library for student use. The general consensus seemed to indicate that the course handouts were helpful to aid understanding of the material. Many students stated they wished there were more handouts and maybe even copies of important sections from the manuals as the material is presented in class. A few students stated they felt a step by step tutorial and/or primer for each application package would be helpful.

Some students commented that the school should advise students prior to their arrival at AFIT of the need for a home computer system. In addition, they felt AFIT should advise the new students about which software packages are used in the graduate program.

Students generally felt that the MathCad portion of the course should have been tied closely to the math review course. They felt that the two courses did not complement each other and the MathCad portion of QMGT 290 should be structured so that the examples and exercises used in the QMGT 290 MathCad section reflect/implement the exercises given in the math review.

The last main area of comments dealt with general maintenance of the systems. Many students felt that having multiple menus on AFIT systems was distracting and at times inhibited them from accomplishing their tasks. That is to say that AFIT should consider a standard menu system for all computers in AFIT so that a person can use any system without having to first figure out how to use that

computer's menu system. For example, room 315 has a generic AFIT built menu system that does not include Wordperfect or Quattro Pro whereas room 312 had the Hard Disk Management (HDM) menu system. Even the systems with HDM aren't all configured the same. Some have all Word Processing programs in a submenu all of its own while others have Wordperfect listed under the Main Programs submenu. This method is confusing to students, especially the beginners. The students strongly feel the systems must be standardized so that the student can concentrate on using the applications, not on finding the software hidden in the menu system. In addition, many students found that many of the systems lacked the appropriate device drivers, screen drivers and utility programs to operate Wordperfect. Lastly, some students feel the school should have a separate overhead projection capability (including screen) for the computer projector as opposed to switching between the overhead slide projector and the computer projector.

On the positive side, the majority of students felt that the course provided a basic foundation and introduction to the software. They realized that there wasn't enough time to provide proficiency with any of the three applications packages but that the objective of the course was to orient the students to the capabilities available to them to complete course work at AFIT. Most students felt the course successfully accomplished its stated objective. Students tended to like the application orientation of the course using actual Air Force examples to demonstrate the capabilities of the software. Most students felt that by showing how to share data between applications, the course promoted the idea of using the software in concert for decision making both in school and on the job.

IV. Conclusions/Recommendations

Significance of Research

The meaning of computer literacy at AFIT is evolving. As the meaning of computer literacy is evolving so must the focus of QMGT290. Prior to this research QMGT290 focused on basic microcomputer orientation to include computer terminology, DOS, VAX VMS, and bulletin boards. This research proposed restructuring the course to adapt an application software orientation. The application software orientation prepares the new student for using the computer assets provided at AFIT during the graduate program. Since there is no other preparatory course provided to develop the computer application prowess necessary to complete the graduate program, this course plays a vital role at AFIT. This research has shown that microcomputers do play an important role in the graduate curriculum. From the first math class to the end of the Thesis, the microcomputer is used to maximize the students' learning experience at AFIT.

This research was intended to answer four research questions:

1. What computer s' ''s are necessary for minimum literacy at AFIT?
2. Which applications software packages are used in the AFIT graduate program?
3. Given the course-time available to teach QMGT 290, what should be taught to maximize the literacy of the incoming students?
4. What teaching method(s) is/are best to teach computer applications?

Throughout the research the author's quest was to develop a revised version of QMGT 290 that better fulfills the needs of incoming AFIT students. The author focused on the microcomputer needs of the new AFIT student. Given the data presented in the Implementation/Analysis section of this document, it appears that the revised QMGT 290 course met these requirements. The course developed minimum

competency among the incoming students by introducing them to three major applications packages, Mathcad™, Quattro Pro™, and WordPerfect™. Based on comments from the students who attended the revised course and the faculty who taught the revised course, it has been determined that the revised course did accomplish its objective of developing a minimum competency in the applications mentioned above. The mix of teaching methods, lecture, hands-on, and take-home exercises appear to have been an acceptable approach to the subject matter based on the comments of students and instructors.

Even though the revised course appears to have been successful, there are areas which require further improvement. If the course continues to focus on the three software packages, additional in-class time may be required to develop a more in-depth understanding of each of the packages. The course developed by this research was limited in depth due to time constraints in the summer short term. This limitation caused the author to take a narrow view of the needs of the students. Each application software program was restricted to a one or two day cursory exposure. In order to elevate the new student competency to a level higher than minimum proficiency, classes will need to be expanded to allow more complete coverage of the software. Several areas must be addressed to bring this expansion to fruition: graduate classroom computer resources must be reallocated to permit additional classroom time. Section size and composition must be re-examined to determine the best scheduling method for future classes to maximize learning potential. As mentioned earlier, a testing mechanism needs to be developed to determine placement of students in sections based on computer aptitude. This test must address each software package separately. The test(s) may need to be evaluated independently to place students into appropriate sections for each software package. Course modules will need to be developed for each level of competency targeted.

This research has developed an approach to address application oriented computer training at AFIT. It provided a basis for the revised QMGT 290 presented for the first time in the summer of 1990. This author has provided data for the AFIT faculty to refine the QMGT 290 course in the future. This refinement should be an iterative process of continually modifying the course each year to address the changing needs of the incoming class. This author believes that computer proficiency levels are dynamically changing with each new class. As each class becomes more computer literate upon arrival at AFIT, the course must evolve into a more advanced applications course instead of the basic level course it is today.

Recommendations for Future Research

This author can see great possibilities for future research in the computer literacy arena. Since the literacy levels of incoming students seems to be increasing with each new class, the possibilities are limitless. Some potential areas of interest include:

- Developing a testing mechanism to segregate students into proficiency levels.
- Developing a survey mechanism to determine computer literacy trends in new classes.
- Developing core modules for the revised QMGT290 that address the different proficiency levels discussed herein.

As mentioned in Chapter III herein, if a module approach is to be successfully implemented with sections differentiated by proficiency levels, then an appropriate testing mechanism must be developed to determine placement of students into these sections. Before such a test could be developed though, a clear definition of expected competency associated with each level of proficiency must be developed. This author recommends Hansgate's course structure presented in Chapter II as a starting point. The outline provides the elements that should be included in each level of a three level

course. This outline would provide the basis to build the testing mechanism. The test should determine the student's ability to complete each of the areas included in the outline.

Determining trends in new student computer literacy could help the course director determine the needs of each incoming class. Since the needs of each new class appear to be changing with the rapidly changing computer industry and the steady influx of microcomputers into the workplace, this survey mechanism should be designed to assist the course director to modify this course each year. The mechanism should determine specific computer competencies, including system familiarization, computer terminology, level of use of the computer in the workplace, the awareness of the manager of the usefulness of the microcomputer in the workplace, etc. Capt Richard Lenz prepared a literacy test in his masters thesis that could be used as a springboard to propel the researcher toward a tool that could be used each year by the course director.

As this course develops into a more in-depth course of study, modules must be developed to implement the course at different competency levels. These modules should be developed to enable students to attend a section best suited for his/her computer applications aptitude. Again, Hansgate's course outline provides the starting point for this activity. The future research may desire to develop modules for each applications software package independently so that students could have the flexibility to attend the level of each application package that best fits his/her abilities.

Conclusions

The world as we know it is changing daily. The United States is battling massive government debt that threatens to burden not only today's adults but also tomorrow's children. This reality is the catalyst for recent congressional and administration activities toward massive cutbacks in government spending to reduce

the burden we leave behind for our children. With the inevitability of reduced budgets, the government manager will be required to do more with less. Individual and organizational productivity will be the watchwords of the 21st century. Computers will become more important as the manager makes use of the computer to accomplish many of his/her daily decision making activities. Computer literacy will take on new meaning in this new world; it will truly be a necessity and not a luxury. Training must fill this necessity today to prepare for tomorrow.

The mission of AFIT "is to provide education to meet Air Force requirements in scientific, technological, managerial, medical, and other fields as directed by HQ USAF" (1:2). An important part of AFIT's mission is to provide trained personnel to fill positions requiring decision making skills. These skills must include computer literacy. Now, more than ever, with so many computers entering the office each year, computer literacy is required for the modern decision maker. If AFIT is to develop this literacy during the summer short term, the faculty will be required to continue to address the application software used in the decision maker's environment. Only with this commitment, on the part of the faculty and the students, will the Air Force gain a computer literate manager from AFIT.

Appendix A: MRDOS DOS Tutorial

During your tour at AFIT you will find that the computer will become part of your everyday life. Therefore it is important to learn how to manage your computer. This tutorial should develop the basic groundwork to accomplish this task. You will NOT be taught the basics of how to use microcomputers or DOS at AFIT. The introductory course given during the summer short term will be application oriented. We will assume you are already competent with DOS and basic computer operation upon arrival at AFIT. You are strongly encouraged to complete the MRDOS tutorial and obtain a competent MS-DOS reference text.

What MRDOS is.....

MRDOS is a disk based tutorial for Microsoft's Disk Operating System (MS-DOS). It provides a basic introduction on using DOS commands to manage your MS-DOS files and directories. This tutorial provides you with the necessary beginning knowledge level of DOS required at AFIT.

What MRDOS is not.....

MRDOS is not a substitute for a good MS-DOS reference manual. It is NOT intended to replace a reference; it is intended to supplement a reference.

MRDOS is not free. MRDOS is part of an inexpensive software source for the IBM-compatible computers called shareware. Shareware is intended to provide a low cost alternative to some commercial software. You are not required to register your copy of MRDOS unless you use it. However, if you find MRDOS useful, you are requested to register your copy with the software author (listed on the opening screen of the program). If you do not intend to use the program, disregard registering the software.

AFIT does not sell, or otherwise have any interest in this program. It is to be interpreted only as a signal that AFIT recognizes solutions come from many sources.

How to use MRDOS.....

The MRDOS program resides on the enclosed diskette. To use MRDOS:

1. After starting your computer, place your MRDOS diskette into your floppy disk drive (A:>).

2. If the current drive is other than A:>, type A: at the system prompt and press RETURN (ENTER).

3. Type MRDOS and press RETURN (ENTER).

4. This will start the tutorial. From this point follow the instructions given in the tutorial.

Have fun and good luck on learning DOS. If you should experience any problems with this tutorial, please call Capt Steve Heaps, AFIT/LSG, AV785-5435.

Appendix B: Course Syllabus

QMGT 290

I. COURSE DESCRIPTION

This course introduces students to AFIT computer systems and the applications software that will be used during graduate studies. The focus is on MS-DOS operating system based microcomputers and to some extent VSM operating system based minicomputers. The course is tailored to prepare students to use three major computer tools available at AFIT for coursework. Since this course is an applications oriented course, the discussions will center on three commercial software packages: Mathcad (mathematical/statistical analysis), Quattro Pro (spreadsheet analysis), and WordPerfect (word processing). The course will be conducted as a combination of classroom and laboratory, with emphasis on hands-on learning.

II. COURSE OBJECTIVES

At the completion of the course, each student will be able to:

A. Install, start, and use Mathcad

B. Install, start, and use Quattro Pro

C. Install, start, and use WordPerfect

D. Demonstrate ability to share files between the above software

E. Demonstrate the ability to transfer files via Electronic Mail.

III. REQUIRED COURSE MATERIALS

A. Mathcad Software. Note this software will also be used in the math review during the Summer short term and well as in follow-on statistics courses.

B. Quattro Pro Software. Note this software will also be used in LOGM 510 and can be applied to many other quantitative courses throughout the graduate program.

C. WordPerfect Software. Although this software is not directly required for any other course, it will become invaluable in thesis work as well as individual course writing assignments.

D. Other materials distributed in class by the instructor.

IV. COURSE EVALUATION CRITERIA

This course is a Pass/Fail course, graded as either Satisfactory or Unsatisfactory. The grade will be determined by your class participation, as well as your performance on four assigned lab projects. You must satisfactorily complete all four projects to receive a passing course grade.

V. COURSE SYLLABUS

WEEK 1 - Mathcad

Day 1 - Course Introduction. Introduction to Mathcad.

Day 2 - Mathcad applications

WEEK 2 - Mathcad/Quattro Pro

Day 1 - Wrap-Up of Mathcad

Day 2 - Getting Started with Quattro Pro

WEEK 3 - Quattro Pro/WordPerfect

Day 1 - Spreadsheet analysis

Day 2 - Getting Started with WordPerfect

WEEK 4 - WordPerfect

Day 1 - Using WordPerfect

Day 2 - Putting it all together with WordPerfect/Review/Wrap-up

INTRODUCTION TO AFIT COMPUTERS

DESCRIPTION

OBJECTIVES

TEXTS

GRADE DETERMINATION

PROPOSED SCHEDULE

INTRODUCTION TO MATHCAD

INTRO1

DESCRIPTION

TITLE:	INTRODUCTION TO AFIT COMPUTERS
NUMBER:	QMGT 290
HOURS:	5, UNDERGRADUATE
INSTRUCTOR:	RICHARD E. PESCHKE, Ph.D., LtCol
OFFICE:	ROOM 101E, BLDG 641
OFFICE HOURS:	1000-1300 Tue;1000-1200 Thu
EMAIL ADDRESS:	RPESCHKE

THE COMPUTER TECHNOLOGIES OF TODAY AND THE NEAR FUTURE WILL GIVE THE MANAGER THE CAPABILITY TO WORK WITH KNOWLEDGE AS NEVER BEFORE.

THIS COURSE INTRODUCES THE COMPUTER SYSTEMS AND MAJOR APPLICATION SOFTWARE PROGRAMS USED AT AFIT.

EMPHASIS OF THIS COURSE IS BREADTH – NOT DEPTH – OF KNOWLEDGE

INTRO2

OBJECTIVES

... BE ABLE TO ...

- INSTALL, START, AND USE MATHCAD
- INSTALL, START, AND USE QUATTRO PRO
- INSTALL, START, AND USE WORDPERFECT

INTRO3

LEARNING PARADIGM

A PARADIGM OF LIFELONG LEARNING SHOULD:

- **REQUIRE RIGOROUS MASTERY OF THE SUBJECT MATTER.**
- **HONE THE CONCEPTUAL SKILLS THAT GENERATE INFORMATION FROM DATA**
- **PROMOTE A HEALTHY SKEPTICISM THAT TESTS REALITY FROM A NUMBER OF PERSPECTIVES.**
- **NOURISH INDIVIDUAL CREATIVITY AND ENCOURAGE EXPLORATION.**
- **SUPPORT COLLABORATION**
- **PROVOKE A JOURNEY OF DISCOVERY**
- **BE ENERGIZED BY THE OPPORTUNITY TO CONTRIBUTE TO THE WHOLE, THAT WHICH WE KNOW AND CAN DO.**

INTRO4

TEXTS

MATHCAD Software Program, Mathsoft, Inc.

QUATTRO-PRO Software Program, Borland, International.

WORDPERFECT Software Program, WordPerfect Corp.

ADDITIONAL HANDOUTS PREPARED BY THE INSTRUCTOR

ANYTHING ELSE THAT YOU FEEL RELATES TO THE SUBJECT, AND THAT WILL AID YOUR LEARNING EXPERIENCE

INTRO5

GRADE

DETERMINATION

YOUR GRADE WILL BE BASED ON YOUREF FORTS IN TWO AREAS:

THREE GROUP PROJECTS

IN-CLASS PARTICIPATION

INTRO6

PROPOSED SCHEDULE

DAY 1: INTRODUCTION TO QMGT 290/INTRODUCTION TO MATHCAD

DAY 2: MATHCAD USE/ PROJECT 1 ASSIGNED

DAY 3: WRAP-UP MATHCAD

DAY 4: INTRODUCTION TO SPREADSHEETS/QUATTRO PRO

DAY 5: USING QUATTRO PRO/ PROJECT 2 ASSIGNED

DAY 6: INTRODUCTION TO WORD PROCESSING/WORDPERFECT/
PROJECT 3 ASSIGNED

DAY 7: USING WORDPERFECT

DAY 8: HOLIDAY - NO CLASS

INTRO7

MATHCAD
WHAT IS IT?????

"MATHCAD IS A UNIQUE NEW WAY TO DEAL WITH MATH. IT IS AS VERSATILE AS THE MOST ADVANCED COMPUTATIONAL TOOLS AND PROGRAMMING LANGUAGES, YET IT IS AS EASY AS A TEXT EDITOR"

MATHCAD BRINGS TRUE WHAT YOU SEE IS WHAT YOU GET CAPABILITY TO ELECTRONIC MATH.....

MC1

MATHCAD
WHAT YOU NEED

- IBM PC, XT, AT, PS2, OR COMPATIBLE
- MS(PC)-DOS 2.0 OR HIGHER
- VGA/EGA/CGA/HERCULES MONITOR AND CARD
- AT LEAST 512 K RAM
- AT LEAST ONE DOUBLE-SIDED 5 1/4" OR 3 1/2"
- DISK DRIVE (HARD-DISK DRIVE OPTIONAL)
- MATH COPROCESSOR (8087/80287/80387) OPTIONAL

MC2

MATHCAD
PRINTER SUPPORT

- IBM PROPRINTER, IBM GRAPHICS PRINTER
- EPSON FX,JX,EX,LX,LQ SERIES PRINTERS
- HP THINKJET/QUIETJET/LASER JET SERIES
- TOSHIBA P351
- OKIDATA 90 SERIES
- C. ITOH C-315
- NEC P5 PINWRITER
- ALL PLOTTERS USING HPGL
- OR COMPATIBLES OF THE ABOVE

MC3

MATHCAD
INSTALLATION

- WHAT YOU NEED:
  - MATHCAD SYSTEM AND AUXILLARY DISKS
  - DOS *
  - ONE OR MORE BLANK DISKS

* DOS USED TO BOOT COMPUTER AND MAKE BACKUP COPY OF MATHCAD

MC4

MATHCAD
INSTALLATION (CONT'D)

- BOOT COMPUTER
- MAKE BACKUP OF MASTER DISKS
  - SINGLE DRIVE SYSTEMS: DISKCOPY A:
  - DUAL DRIVE SYSTEMS (SAME SIZE DRIVE): DISKCOPY A: B:
- FOR NON-HARD DISK SYSTEMS, THIS CONCLUDES THE INSTALLATION

MC5

MATHCAD
INSTALLATION (CONT'D)

- HARD DISK SYSTEMS
  - MAKE A MATHCAD SUBDIRECTORY ON THE HARD DRIVE (MD MCAD)
  - COPY THE BACKUP SYSTEM DISK INTO THE MCAD SUBDIRECTORY

NOTE: THE AUXILLARY DISK CONTAINS SAMPLE TEMPLATES. IT IS NOT NECESSARY TO COPY THESE ON THE HARD DRIVE. IF THE USER DESIRES TO COPY THIS DISK ONTO THE HARD DRIVE, REPEAT STEP 5 WITH THE AUXILLARY DISK.

- THIS CONCLUDES INSTALLATION FOR HARD DISK

MC6

MATHCAD
STARTING A SESSION

NON HARD DISK USERS:

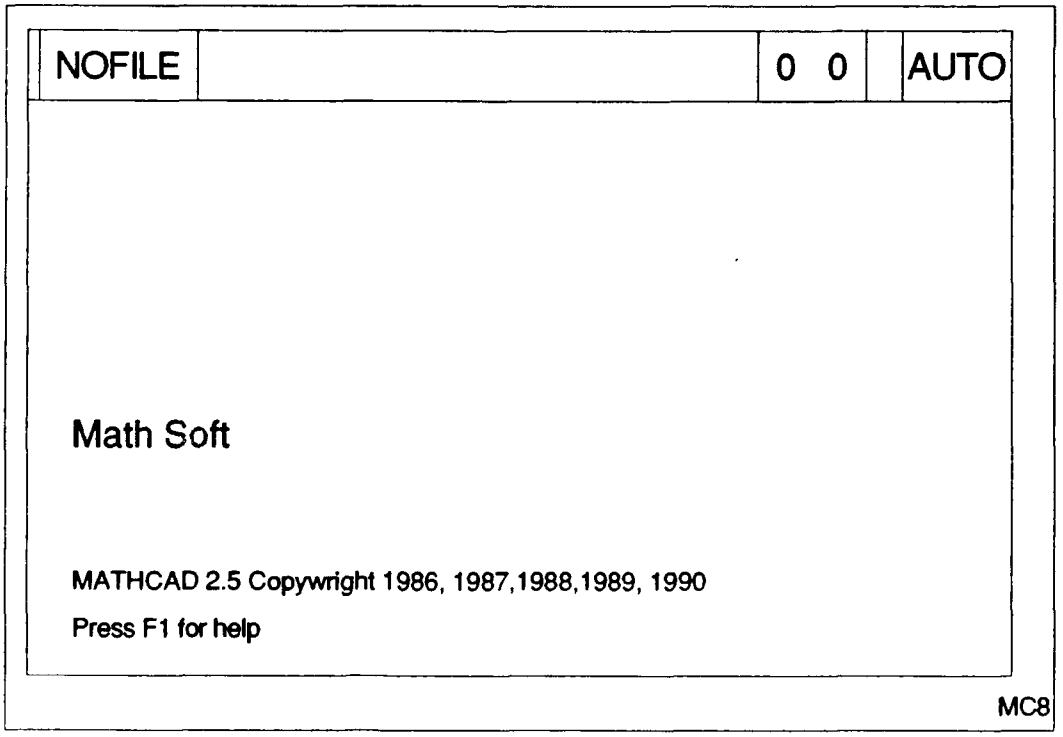
1. PUT MATHCAD SYSTEM DISK IN THE "A" DRIVE
2. TYPE MCAD

HARD DISK USERS:

1. MAKE THE MCAD SUBDIRECTORY CURRENT

2. TYPE MCAD

MC7



MC8

MATHCAD
FIRST PRINCIPLES

- EVERYTHING APPEARS IN FAMILIAR NOTATION
- WHAT YOU SEE IS WHAT YOU GET (WYSIWYG)
- TO CREATE SIMPLE EXPRESSIONS, JUST TYPE THEM IN
- TO CREATE COMPLICATED CONSTRUCTS, JUST FILL IN THE BLANKS
- CALCULATION FEATURES ARE MODULAR
- COMMANDS ARE AVAILABLE ON KEYS, BY NAME, BY MENU
- NUMERICAL ALGORITHMS ARE ROBUST, STANDARD, PREDICTABLE
- EVERYTHING IS DOCUMENTED

MC9

MATHCAD
MESSAGE LINE

- REVERSE VIDEO AREA AT TOP OF SCREEN
- DISPLAYS
  - CURRENT FILENAME
  - MESSAGES
  - CURSOR POSITION INDICATOR
  - CALCULATION MODE
    - AUTO(MATIC)
    - MAN(UAL)

MC10

MATHCAD
REGIONS

- ▶EQUATION
  - ▶MATHEMATICAL CALCULATIONS
  - ▶DEFAULT
  - ▶Y = mX + b
- ▶PLOT
  - ▶GRAPHIC DISPLAYS
  - ▶RELATED TO EQUATION REGION
  - ▶@
- ▶TEXT
  - ▶DOCUMENTATION REGION
  - ▶"

MC11

MATHCAD
EQUATIONS

- ▶PARTS
  - ▶NUMBERS
  - ▶VARIABLES
    - ▶START WITH LTR
    - ▶CAN BE ALL LTRS, OR COMBO OF LTRS & NUMBERS OR CERTAIN GREEK LTRS
  - ▶OPERATORS
    - ▶+,-,*,/,\,>,<,^
  - ▶PARENTHESES
    - ▶SHOW ORDER OF CALCULATION
    - ▶ARGUMENT OF A FUNCTION

MC12

MATHCAD
KINDS OF EQUATIONS

- ▶CALCULATION
  - ▶DISPLAYS RESULT OF A CALCULATION
  - ▶=
- ▶DEFINITION
  - ▶2 TYPES
    - ▶GLOBAL
      - ▶EFFECTIVE IN ENTIRE WORKSHEET
      - ▶~
    - ▶LOCAL
      - ▶DEFINED FROM LOCATION IN TEMPLATE FORWARD
      - ▶:

MC13

MATHCAD
EQUATIONS

CALCULATION	DEFINITION
9+5	X:=125/5
6-5	Y:=(187-5)/2
8*7	Y:=187-(5/2)
21/3	Z:=X+Y
$\sqrt{2}$	$R:=Y^{(X/5)}$
9^2	$M:=\sqrt{X}$

MC14

MATHCAD
FUNCTION KEYS

F1	F2
F3	F4
F5	F6
F7	F8
F9	F10

F1: HELP
F2: COPY
F3: CUT
F4: PASTE
F5: LOAD
F6: SAVE
F7: SPLIT
F8: SWITCH
F9: CALCULATE
F10: COMMAND MENUS

MC15

MATHCAD
CTRL + FUNCTION KEY

F1	F2
F3	F4
F5	F6
F7	F8
F9	F10

CTRL F1: NO DEFINITION
CTRL F2: INCOPY
CTRL F3: INCUT
CTRL F4: INPASTE
CTRL F5: SEARCH
CTRL F6: REPLACE
CTRL F7: UNSPLIT
CTRL F8: NO DEFINITION
CTRL F9: INSERT LINE
CTRL F10: DELELE LINE

MC16

MATHCAD
CURSOR KEYS

↑↓ → ←	MOVE ONE LINE OR COLUMN IN SPECIFIED DIRECTION
TAB	IN TEXT: MOVE TO NEXT WORD IN EQUATION/PLOT: MOVE TO NEXT PLACEHOLDER BETWEEN REGIONS: MOVE RIGHT 10-CHAR
SHIFT TAB	IN TEXT: MOVE BACK ONE WORD IN EQUATION OR PLOT: MOVE TO PREVIOUS PLACEHOLDER BETWEEN REGIONS: MOVE LEFT 10 CHAR
PgUp	MOVE UP 5 LINES AT A TIME
PgDn	MOVE DOWN 5 LINES AT A TIME
Ctrl PgUp	MOVE UP 80% OF PAGE
Ctrl PgDn	MOVE DOWN 80% OF PAGE

MC17

MATHCAD
CURSOR KEYS (CONT'D)

Ctrl ←	MOVES LEFT TO NEXT 10 CHAR TAB
Ctrl →	MOVES RIGHT TO NEXT 10 CHAR TAB
HOME	MOVES TO BEGINNING OF CURRENT REGION. IF ALREADY THERE, MOVES TO PREVIOUS REGION
END	MOVES TO END OF CURRENT REGION. IF ALREADY THERE, MOVES TO NEXT REGION
Ctrl HOME	SCROLL TO BEGINNING OF DOCUMENT
Ctrl END	SCROLL TO END OF DOCUMENT

MC18

MATHCAD
PRACTICE EXERCISES

PROBLEM 1: DETERMINE THE LENGTH OF THE HYPOTENUSE OF A RIGHT TRIANGLE WITH A BASE = 27 INCHES AND A HEIGHT = 36 INCHES.

PROBLEM 2: DETERMINE THE AREA ENCLOSED BY THE TRIANGLE ABOVE.

PROBLEM 3: PLOT THE TRIANGLE ABOVE.

MC19

MATHCAD
ENDING A SESSION

3 OPTIONS

1 F-10
SYSTEM
QUIT
- OR -

2 ESC Q(UIT)
- OR -

3 CTRL-Q

NOTE: IF TEMPLATE HAS NOT BEEN SAVED OR CHANGES HAVE BEEN MADE, MATHCAD WILL ADVISE YOU THAT YOU WILL LOSE YOUR TEMPLATE/CHANGES IF YOU DO NOT RESAVE IT PRIOR TO EXITING FROM MATHCAD

MC20

INTRODUCTION TO SPREADSHEETS

WHAT IS A SPREADSHEET?

WHEN/WHERE USE IT?

WHO SHOULD USE IT?

WHY USE IT?

INTRODUCTION TO QUATTRO PRO

QPRO1

WHAT IS A SPREADSHEET

AN ALL PURPOSE TOOL

AN ELECTRONIC VERSION OF A LEDGER BOOK

A SPREADSHEET IS TO NUMBERS WHAT A WORD PROCESSOR IS TO WORDS

IT RECORDS FIGURES AND OTHER INFORMATION

MOST COMMON APPLICATION - BUDGETING BUT TODAY'S SPREADSHEETS ARE MORE THAN JUST BUDGETING TOOLS

QPRO2

WHEN/WHERE TO USE A SPREADSHEET

BUDGET/CHECKBOOK REGISTER

FINANCIAL ANALYSIS

CHECKLISTS

SCHEDULE MAKING

LINEAR PROGRAMMING

REGRESSION

PRODUCTION MANAGEMENT CALCULATIONS

QPRO3

WHO SHOULD USE A SPREADSHEET

ANYONE WHO WORKS WITH INFORMATION

FINANCIAL MANAGERS

PRODUCTION MANAGERS

PROJECT MANAGERS

. . . STUDENTS . . .

AND MANY MORE

QPRO4

WHY USE A SPREADSHEET

"WHAT-IF" ANALYSIS

EASE OF CHANGING VALUES

REPEATABILITY

QUALITY APPEARANCE

EASIEST COMPUTER APPLICATION TO LEARN & USE

QPRO5

KEY FEATURES OF QUATTRO PRO

SIZE -- 8192 ROWS BY 256 COLUMNS -- 2097152 CELLS

LINKING -- LINK TOGETHER SPREADSHEETS OR GRAPHICS

WINDOWS -- UP TO 32 WINDOWS OPEN AT ONCE

UNDO CAPABILITY

LINEAR PROGRAMMING

REGRESSION

SUPERB GRAPHICS WITH BUILT IN ANNOTATE

DIRECTLY COMPATIBLE WITH WORDPERFECT

QPRO6

FUNDAMENTALS OF QUATTRO PRO

WORKAREA ORIENTATION

COMMAND MENU

BASIC ENTRY

PRINTING

BASIC FUNCTIONS

FILE MANAGEMENT

GRAPHICS

QPRO7

BASIC SPREADSHEET CONCEPTS

- WORKSHEET: WORK AREA IN SPREADSHEET; WHERE INPUT GOES
- CELL
  - MOST BASIC ELEMENT OF SPREADSHEET
  - DEFINED BY INTERSECTION OF COLUMN AND ROW
  - ADDRESS = COLUMN-ROW LABEL OF CELL (E.G. A1 IS ADDRESS OF CELL AT INTERSECTION OF COLUMN "A" AND ROW "1"
- BLOCK:
  - GROUP OF CONTIGUOUS CELLS IN A SPREADSHEET; CAN BE
    - ONE CELL
    - SINGLE ROW OF CELLS
    - SINGLE COLUMN OF CELLS
      OR A GROUP OF ROWS AND COLUMNS (CONTIGUOUS)

QPRO8

BASIC SPREADSHEET CONCEPTS (CONT'D)

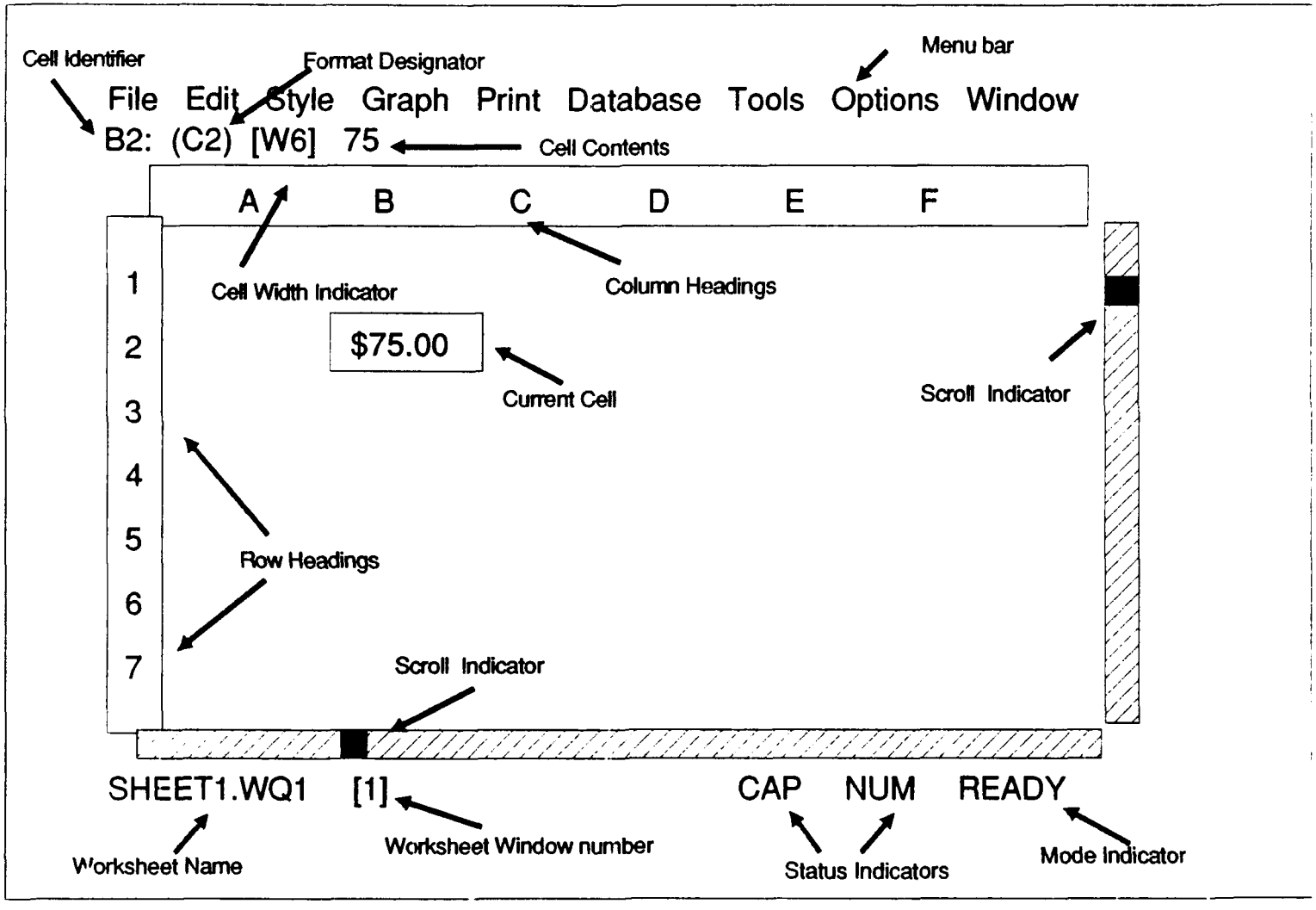
- LABEL: SEQUENCE OF CHARACTERS (LETTERS, DIGITS, OR PUNCUTATION MARKS) USED TO ADD VERBAL DESCRIPTION
- NUMERIC: CELL ENTRY THAT IS A NUMBER
- FORMULA: INSTRUCTIONS TO QPRO TO DO A CALCULATION
- INPUT MODE: USED TO ENTER DATA OR MOVE CURSOR
- COMMAND MODE: USED TO PERFORM OPERATIONS ON DATA CURRENTLY IN THE WORKSHEET OR ON DISK.
- FUNCTIONS: A BUILT-IN FORMULA; USUALLY STARTED WITH @

QPRO9

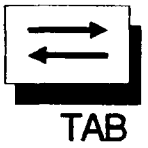
BASIC SPREADSHEET CONCEPTS (CONT'D)

- COPYING
  - SOURCE BLOCK: WHAT TO COPY (CAN BE ONE OR MORE CONTIGUOUS CELLS)
  - DESTINATION BLOCK: WHERE TO PUT THE RESULTS OF THE COPY COMMAND (CAN BE ONE OR MORE CONTIGUOUS CELLS)
- CELL ADDRESSING
  - RELATIVE: A CELL ADDRESS THAT IS ADJUSTED WHEN IT IS COPIED TO A NEW LOCATION; NOT LITERAL
    E.G. COPY THE CELL B1 CONTENTS (A1*1.5) TO C2
    WHEN COPIED THE NEW FORMULA IN C2 IS B2*1.5; THE COLUMN AND ROW IS ADJUSTED.
  - ABSOLUTE: LITERAL CELL ADDRESS; WHEN COPIED, THE COPY IS LITERALLY THE SAME AS THE SOURCE
    E.G. COPY CELL B1 AS ABOVE TO C2; THE CONTENTS OF C2 IS THE SAME AS THE CONTENTS OF B1

QPRO10


Cell Identifier
Format Designator
Menu bar
File Edit Style Graph Print Database Tools Options Window
B2: (C2) [W6] 75
Cell Contents
A B C D E F
1 2 3 4 5 6 7
Cell Width Indicator
Column Headings
$75.00
Current Cell
Scroll Indicator
Row Headings
Scroll Indicator
SHEET1.WQ1 [1]
CAP NUM READY
Worksheet Name
Worksheet Window number
Status Indicators
Mode Indicator

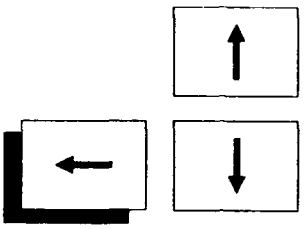
QPRO CURSOR CONTROL



TAB

INSERT | HOME | PAGE UP

DELETE | END | PAGE DOWN



CTRL-LEFT CTRL-RIGHT

QPRO11

FUNCTION KEYS

F1 HELP	F2 EDIT DEBUG MACRO MEN
F3 CHOICES MACROS FUNCTION	F4 ABSOLUTE
F5 GoTo PICK WIN UNDO	F6 PANE NEXT WIN ZOOM
F7 QUERY SELECT ALL SELECT	F8 TABLE MOVE
F9 CALC COPY	F10 GRAPH PASTE

QPRO12

TYPES OF DATA

TWO TYPES

VALUES

#
CALCULATIONS (+ - * / ^)
FORMULAS (@)

LABELS

WORDS (' ^ ")

ENTER VS ARROW KEYS

QPRO13

EDITING DATA

TYPEOVER

F2 (EDIT)

BS
ARROWS
DEL

DEL

QPRO14

CHANGE YOUR MIND???

ESC KEY

CLEARS NEW DATA FROM INPUT LINE
(BEFORE PRESSING RETURN)

MOVES BACK ONE STEP IN MENU

CTRL+BREAK

MOVES YOU COMPLETELY OUT OF MENU

QPRO15

SAVING AND RETRIEVING DATA

SAVE

NEW, UNSAVED DATA

/File Save {filename}

PREVIOUSLY SAVED DATA

/File Save {filename} Replace

RETRIEVE

/File Retrieve {filename}

QPRO16

PRINTING IN QUATTRO PRO

- ▶ QPRO CAN PRINT
  - ▶ SIMPLE(BASIC), SINGLE PAGE PRINTOUTS
  - ▶ COMPLEX, HIGHLY FORMATED, MULTIPAGE PRINTOUTS
  - ▶ SPREADSHEET DATA TO A TEXT FILE (CAN BE IMPORTED INTO)
    - ▶ MATHCAD
    - ▶ WORDPERFECT/ OTHER WORD PROCESSORS (ASCII)
    - ▶ STATISTIX
- ▶ BASIC BLOCK PRINTING (3 COMMANDS)

  /PRINT, BLOCK, SPREADSHEET PRINT

  /PRINT: INITIATES THE PRINT PROCESS

  BLOCK: DESIGNATES WHAT YOU WANT PRINTED

  SPREADSHEET PRINT: STARTS THE PRINTING*

*NOTE: BE SURE PRINTER IS PLUGGED IN, TURNED ON AND LOADED WITH PAPER.

QPRO17

PRINTING IN QUATTRO PRO

- ▶ COMPLEX, HIGHLY FORMATED, MULTIPAGE PRINTOUTS
  - ▶ CAN CHANGE
    - ▶ HEADERS/FOOTERS (/PLH or F)
    - ▶ CHANGE MARGINS (LEFT, RIGHT, TOP, BOTTOM) (/PLM)
    - ▶ PAGE SIZE (/PLD)
    - ▶ ORIENTATION (PORTRAIT/LANDSCAPE) (/PLO)
    - ▶ FORMAT (AS DISPLAYED/CELL FORMULAS) (/PF)
    - ▶ ADJUST PRINTER (SKIP LINE, FORM FEED, ALIGN) (/PA)
    - ▶ DESTINATION (PRINTER, FILE) (/PD)
  - ▶ SPREADSHEET DATA TO A TEXT FILE

    1) SPECIFY BLOCK TO BE EXPORTED (/PB)

    2) CHANGE MARGINS TO "0" ALL AROUND (LM)

    3) SELECT FILE FROM DRAFT MODE PRINTING (DF)

    4) SPECIFY NAME FOR FILE (QPRO WILL SUPPLY .PRN EXTENSION)

    5) SELECT SPREADSHEET PRINT (S)

QPRO18

GRAPHS IN QPRO

BASIC CONCEPTS

- GRAPH TYPES
  - LINE
  - BAR
  - XY
  - STACKED BAR
  - PIE
  - AREA
  - ROTATED BAR
  - COLUMN
  - HIGH-LOW
  - TEXT
- SERIES
  - DATA TO BE GRAPHED
  - CAN HAVE MAX OF 6
  - X-AXIS SERIES ARE VALUES ALONG X-AXIS
- TITLES
  - GRAPH TITLE/SUBTITLE
  - X-AXIS TITLE
  - Y-AXIS TITLE
  - LEGENDS

QGRAPH

BASIC MECHANICS OF GRAPHS

BUILDING

SELECT /Graph FROM MENU

SELECT Graph Type AND DESIGNATE THE TYPE DESIRED

SELECT Series AND DESIGNATE EACH SERIES

SELECT Titles AND DESIGNATE ALL TITLES

SELECT View TO PREVIEW THE GRAPH

PRINTING

TO PRINTER: SELECT /Print, Graph print

TO FILE:

SELECT /Print, Write EPS/PIC File, PIC

INPUT .PIC filename

QGRAPH1

EXITING QPRO

THROUGH MENU SYSTEM

/File Exit

IF THE DATA ON THE SCREEN AT THE TIME THIS COMMAND IS ISSUED HAS NOT BEEN SAVED, QPRO WILL RESPOND WITH:

LOSE YOUR CHANGES AND EXIT?

SHORTCUT:

CTRL + X

IMMEDIATELY EXITS FROM QPRO (IF DATA NOT SAVED, QPRO RESPONDS WITH SAVE DIALOGUE AS ABOVE)

QPRO19

FUNDAMENTALS OF WORDPERFECT 5.0/5.1

- PURPOSE
- KEYBOARD LAYOUT
- CURSOR CONTROL
- COMMAND MENU
- TEXT FORMATTING
- TEXT ENHANCEMENTS
- GRAPHIC ENHANCEMENTS
- FILE MANAGEMENT
- PRINTING

WP0

What is Wordperfect?

- A WORD PROCESSOR
  - COLLECTION OF FUNCTIONS WHICH ALLOWS STORING, EDITING, AND MANIPULATING OF TEXT USING A COMPUTER
  - AUTOMATED TYPEWRITER
- A MID LEVEL DESKTOP PUBLISHER
  - INTEGRATES TEXT AND GRAPHICS
  - INCLUDES UTILITIES THAT DO TYPESETTING-TYPE FUNCTIONS

WP1

COMMON FEATURES OF WORD PROCESSORS

- TEXT EDITING
- WORD WRAP
- TEXT JUSTIFICATION
- CUT & PASTE
- MOVE & COPY
- FIND & REPLACE
- PAGE LAYOUT SETTINGS

WP2

INSTALLATION REQUIREMENTS

- WORDPERFECT PROGRAM DISKS
- DOS 2.0+
- 384k RAM(MIN)
- CGA/EGA/VGA/MONOCHROME MONITOR

WP3

BASIC INSTALLATION

- PUT LEARN DISKETE INTO DOSK DRIVE
- AT THE SYSTEM PROMPT (C:) TYPE A:INSTALL

WP4

STARTING WORDPERFECT

- BLANK SCREEN (NEW DOCUMENT)
  - TYPE WP <RET>
- START IMMEDIATELY EDITING A DOCUMENT
  - WP <FILENAME>

WP5

SCREEN LAYOUT

- CLEAN SHEET OF PAPER
- STATUS LINE AT BOTTOM OF PAGE
  - CURSOR POSITION
    (DOCUMENT,PAGE,LINE,COLUMN)
  - DOCUMENT NAME

WP6

CURSOR MOVEMENT

CHARACTER/1 LINE	← → ↑ ↓
WORD	CTRL ← → ↑ ↓
BEGINNING OF LINE	HOME ←
END OF LINE	HOME → OR END
TOP OF SCREEN	HOME ↑
BOT OF SCREEN	HOME ↓
TOP OF CURRENT PAGE	CTRL HOME ↑
TOP OF PREVIOUS PAGE	PGUP
TOP OF NEXT PAGE	PGDN
BOT CURRENT PAGE	CTRL HOME ↓
TOP OF DOCUMENT	HOME HOME↑
BOT OF DOCUMENT	HOME HOME↓
TO SPECIFIC PAGE	CTRL HOME #
N "UNITS"	ESC + MOVEMENT KEY

WP7

SHIFT/ALT/CTRL

- SHIFT
  - UPPER CASE LETTERS
  - SPECIAL CHARACTERS
  - ACTIVATES WP OPERATIONS WHEN USED IN CONJUNCTION WITH FUNCTION KEYS
  - HOLD DOWN SHIFT KEY & PRESS OTHER KEY
- ALT/CTRL
  - USED IN CONJUNCTION WITH NUMBERS, LETTERS, & FUNCTION KEYS TO DISPLAY CHARACTERS, AND CARRY OUT WP OPERATIONS
  - HOLD DOWN ALT/CTRL KEY & PRESS OTHER KEY

WP8

SPECIAL PURPOSE KEYS

FUNCTION KEYS
- EACH HAS 4 DEFINITIONS
  - ALONE
  - W/ SHIFT
  - W/ CTRL
  - W/ ALT

ESC
- NOT USED TO CANCEL AN OPERATION
- REPEAT KEY

ENTER
- HARD CARRIAGE RETURN
- ENDS A PARAGRAPH

WP9

CURSOR

- BLINKING UNDERLINE OR BLOCK
- POSITION OF NEW TEXT

WP10

WORD WRAP

- WP AUTOMATICALLY KNOWS END OF LINE
- TYPE CONTINUOUSLY UNTIL END OF A PARAGRAPH (NO CR)
- TYPES OF CARRIAGE RETURNS
  - SOFT
    - PLACED BY WP AT END OF EACH LINE IF RETURN KEY NOT ISSUED BY USER
    - ALLOWS WP TO AUTOMATICALLY REFORMAT TEXT AS IT IS EDITED.
  - HARD
    - USER ISSUED (RETURN/ENTER)
    - USES
      - END PARAGRAPH
      - END SHORT LINE
      - INSERTING A BLANK LINE

WP11

INS/TYPEOVER

- TEXT ENTRY MODE
  - INS
    - PLACES NEW TEXT AT POINT OF CURSOR
    - OLD TEXT MOVES TO RIGHT OF NEW
  - TYPEOVER
    - NEW TEXT "TYPES OVER" OLD TEXT
    - OLD TEXT LOST

WP12

HYPHEN

- TWO TYPES
  - SOFT
    - USED TO INSTRUCT WP WHERE TO HYPHENATE A WORD IF IT FALLS PAST END OF LINE
    - " CTRL - "
  - HARD
    - USED FOR USER DIRECTED HYPHEN REGARDLESS OF WHERE IT APPEARS ON A LINE
    - " - "

WP13

CREATING A DOCUMENT

- ENTERING TEXT
- MOVING CURSOR THROUGH TEXT
- USING FUNCTION KEYS
- SELECTING MENU OPTIONS
- USING CANCEL (F1)
- GETTING HELP (F3)
- USING BOTH DOCUMENT WINDOWS (SHIFT + F3)
- PRINTING A DOCUMENT (SHIFT + F7)
- NAMING AND SAVING A DOCUMENT

WP14

PRINTING IN WORDPERFECT

- BLOCK
  - HILITE BLOCK
  - SHIFT F7
- PAGE
  - CURSOR ANYWHERE ON PAGE DESIRED
  - SHIFT+F7
  - P(AGE)
- ENTIRE DOCUMENT
  - SHIFT+ F7
  - F(ULL)

WP15

IMPORTING GRAPHICS INTO WORDPERFECT

- WORDPERFECT CAN IMPORT GRAPHICS INTO WORD PROCESSING DOCUMENTS TO ENHANCE THE FINAL PRODUCT
- TWO TYPES OF GRAPHICS IN WORDPERFECT
  - LINES, BOXES, AND SHADING (AVAILABLE WITHIN WORDPERFECT ITSELF)
  - IMPORTED GRAPHICS FROM POPULAR SOFTWARE PACKAGES

WPGRAPH

WORDPERFECT COMPATIBLE OBJECT ORIENTED GRAPHICS

SOFTWARE	GRAPHIC FORMAT
AUTOCAD	DEFAULT
CCS DESIGNER	HPGL
CHARTMASTER	HPGL
DIAGRAM MASTER	HPGL
DIAGRAPH	HPGL
FREELANCE PLUS	CGM
GENERIC CAD	HPGL
GRAPH-IN-A-BOX	HPGL
GRAPHWRITER	CGM
HARVARD GRAPHICS	HPGL,CGM,EPS
IBM CADAM	HPGL
IBM CATIA	HPGL
IBM CBDS	HPGL
IBM GPG	HPGL
MICROSOFT CHART	HPGL
PICTURE PAK	DEFAULT
PLANPERFECT	CGM
SIGNMASTER	HPGL
SLIDEWRITE PLUS	HPGL, TIFF, PCX
SYMPHONY	DEFAULT
VERSACAD	HPGL

Note: Default under graphic format means that the software is directly compatible wit with Wordperfect. There is no need to use a special export function in these programs All other graphics must be EXPORTED in the format indicated

WPGRAPH1

WORDPERFECT COMPATIBLE BIT-MAPPED IMAGE GRAPHICS

SOFTWARE	GRAPHIC FORMAT
ADOBE ILLUSTRATOR	EPS
BOEING GRAPH	IMG
CIES (COMPUSCAN)	TIFF
DFI HANDY SCANNER	IMG, TIFF
DR HALO II	DEFAULT
ENTERGRAPHICS	IMG, TIFF
GEM PAINT	DEFAULT(IMG)
GEM SCAN	DEFAULT(IMG)
HP GRAPHICS GALLERY	TIFF, PCX
HP SCANNING GALLERY	TIFF, PCX
LOTUS 1-2-3	DEFAULT (PIC)
MACPAINT	DEFAULT
PC PAINT PLUS	DEFAULT
PC PAINTBRUSH	DEFAULT
PROFESSIONAL PLAN	DEFAULT
QUATTRO	PIC
QUATTRO PRO	PIC
SUPERCALC 4	PIC
SYMPHONY	DEFAULT
VP PLANNER	PIC
WINDOWS PAINT	DEFAULT
WORDS & FIGURES	PIC

NOTE: DEFAULT UNDER GRAPHIC FORMAT MEANS THAT THE SOFTWARE'S DEFAULT GRAPHIC FORMAT IS DIRECTLY IMPORTABLE INTO WORDPERFECT. ALL OTHERS MUST BE EXPORTED INTO THE FORMAT INDICATED.

WPGRAPH2

CREATING A GRAPHIC FILE IN QPRO TO IMPORT INTO WP

- QPRO GRAPHS CAN BE IMPORTED DIRECTLY INTO WP
  - QPRO GRAPHIC FILES = BIT-MAPPED IMAGE (PICture)
  - SAVED IN LOTUS 1-2-3 .PIC FORMAT
- MECHANICS
  1) CREATE COMPLETE GRAPH AS USUAL
  2) SELECT P RINT FROM THE MENU
  3) SELECT G RAPHPRINT
  4) SELECT W RITE GRAPH FILE
  5) SELECT P IC FILE
  6) TYPE FILENAME FOR GRAPHIC FILE (NOTE: QPRO WILL AUTOMATICALLY PROVIDE THE .PIC EXTENSION

WPGRAPH3

IMPORTING GRAPHICS THE MECHANICS

- PLACEMENT OF THE GRAPHIC
  - GRAPHIC APPEARS AT THE POINT WHERE THE CURSOR IS PLACED IN THE DOCUMENT IF
    - THERE IS ENOUGH ROOM ON THAT PAGE FOR THE ENTIRE GRAPHIC
    - IF NOT, WORDPERFECT PLACES THE GRAPHIC AT THE TOP OF THE NEXT PAGE
    - NOTICE THAT EVEN IF WORDPERFECT PLACES THE GRAPHIC AT THE TOP OF THE NEXT PAGE, THE GRAPHIC MARKER (CODE THAT IS VISIBLE IN THE REVEAL CODES MODE) WILL BE PLACED AT THE POINT IN THE TEXT WHERE THE CURSOR IS LOCATED AT THE POINT OF IMPORTATION

WPGRAPH4

THE MECHANICS (CONT'D)

- ► TO ACTIVATE THE WORDPERFECT'S GRAPHIC, CAPABILITIES, PRESS ALT+F9
- ► WORDPERFECT REPLIES WITH THE FOLLOWING:

DEFINITION: FIGURE

1 - Filename	
2 - Caption	
3 - Type	Paragraph
4 - Vertical Position	0"
5 - Horizontal Position	Right
6 - Size	3.25" wide x 3.25" high
7 - Wrap Text Around Box	Yes
8 - Edit	

Selection: 0

WPGRAPH5

THE MECHANICS(CONT'D)

1) SELECT 1 OR FILENAME AND PROVIDE THE GRAPHIC FILENAME WITH THE APPROPRIATE EXTENSION.

2) SELECT 2 OR CAPTION TO INCLUDE THE FIGURE NAME. FIGURE NAMES MUST BE CENTERED ON THE GRAPHIC (DEFAULT FOR WP)

3) SELECT 5 OR HORIZONTAL POSITION AND C ENTER TO CENTER THE GRAPHIC ON THE PAGE

4) SELECT 6 OR SIZE TO CHANGE THE SIZE OF THE GRAPHIC IN THE DOCUMENT. SIZE OF GRAPHIC IS DEPENDENT ON SIZE OF TEXT IN THE GRAPHIC. TEXT MUST BE LARGE ENOUGHT TO READ CLEARLY AND REPRODUCE CLEARLY. NORMALLY, MOST GRAPHS SHOULD BE NO SMALLER THAN 6" WIDE BY 4" HIGH. TO SET THE SIZE AT THESE DIMENSIONS SELECT BOTH AND SPECIFY 6 FOR WIDTH AND 4 FOR HEIGHT

5) IF YOU SELECT 8 OR EDIT YOU CAN NOW SEE THE GRAPHIC IN EDIT MODE IN WP. TO ESCAPE FROM EDIT MODE PRESS F7

6) TO EXIT FROM THE GRAPHIC MENU AND RETURN TO YOUR DOCUMENT PRESS F7

YOUR GRAPHIC IS NOW IN WORDPERFECT. YOU WILL SEE A BOX APPEAR ON THE SCREEN WHERE THE GRAPHIC WILL APPEAR. THE BOX WILL NOT SHOW THE GRAPH. TO VIEW THE PAGE WITH THE GRAPHIC, PRESS SHIFT+F7 AND THEN VIEW.

WPGRAPH6

IMPORTING ASCII TEXT INTO WP

- ASCII TEXT = ORDINARY TEXT WITHOUT PRINTER CONTROL CHARACTERS. USED TO SHARE DATA AMONG DIFFERENT PROGRAMS.
- COMMONREASONS TO IMPORT ASCII FILES AT AFIT
  - EDITING DOWNOLADED TEXT FILES FROM THE VAX (SAS OUTPUT, DIRTY PURPLES, ETC)
  - MOVING TEXT FROM OTHER WORD PROCESSORS
  - INPUTTING SPREADSHEET DATA INTO A DOCUMENT

WPGRAPH7

IMPORTING ASCII TEXT (MECHANICS)

- FROM QUATTRO PRO (QPRO)
  - CREATE TEXT FILE IN QPRO

SELECT PRINT

BLOCK (SPECIFY THE BLOCK TO BE

DESTINATION , F ILE (INPUT YOUR

TEXT FILE NAME)

PRINT SPREADSHEET

NOTE: YOU SHOULD NOTICE THE DRIVE LIGHT ILLUMINATE.

- IMPORTING THE DATA (IN WP)

PRESS CTRL+F5 (TEXT IN/OUT)

SELECT 1 OR DOS TEXT

2 OR RETRIEVE (CR/LF TO HRT)*

ENTER FILENAME (SAME AS ABOVE)

*NOTE: USE SELECTION 2 FOR SPREADSHEET AND TABLE DATA; USE 3 FOR PURE ASCII FILES

WPGRAPH8

EXPORTING ASCII TEXT FROM WP

- COMMON REASONS TO EXPORT ASCII FILES AT AFIT
  - UPLOADING TEXT FILES TO THE VAX (SAS FILES, DIRTY PURPLES, ETC)
  - MOVING TEXT FROM WORDPERFECT TO ANOTHER WORD PROCESSOR
  - EXPORTING DATA TO A SPREADSHEET
  - EXPORTING DATA TO MATHCAD
  - EXPORTING DATA TO STATISTIX
- MECHANICS

  CREATE TEXT
  PRESS CTRL+F5
  SELECT 1 OR DOS TEXT
  SELECT 1 OR SAVE
  INPUT FILENAME FOR ASCII FILE

WPGRAPH9

EXITING WORDPERFECT

- F7
- SAVE DOCUMENT? Y/N
- ENTER FILENAME: FILENAME
- EXIT WP? Y/N

WP16

Appendix D: Exercises/Projects

Day 1: Exercise #1

TRIAG.mcd

Problem #1: Determine the length of the Hypothenuse of a right triangle with a base of 27 and a height of 36 inches.

Base := 27
Height := 36

$HYP := \sqrt{BASE^2 + HEIGHT^2}$ $HYP = 45$

Use the Pythagorean Theorem to determine the length of the hypothenuse of the triangle

Problem #2: Determine the area enclosed by the triangle above.

$Area := \frac{1}{2} \cdot Base \cdot Height$ $Area = 486$ (Area of the above triangle)

Problem #3: Plot the triangle given above.

A triangle is a geometric figure that is made up of three straight lines. In MathCad, to plot a triangle we must plot the three lines separately. We know that the equation of a line is $Y = mX + b$ where m is the slope of the line and b is the intercept of the line.

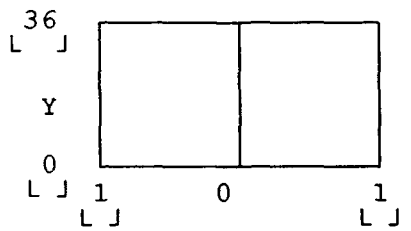
We will start by orienting the right angle of the triangle at the origin. Then we can add each line that makes up the triangle.

Vertical Segment -->

Since the vertical segment starts at the origin and goes up the Y-axis to the value of the height, we know that for all values of Y, X = 0, so:

b := Height (Y intercept)

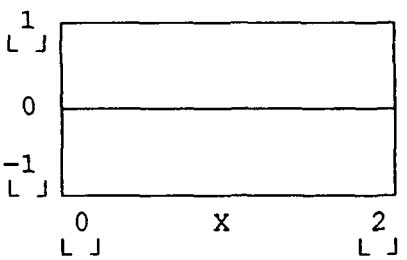
To define this line in MathCad we have Y := 0 ..Height



<-- Notice the vertical line through the origin.

To define the horizontal line we have a slope of 0 and a Y-intercept of 0, therefore, the line goes thru the origin to the point (Base,0)

X := 0 ..Base (for all values of X, Y = 0)



<-- Notice the horizontal line through the origin to (27,0)

Finally, we plot the diagonal line. Here we have both a slope and an intercept as follows:

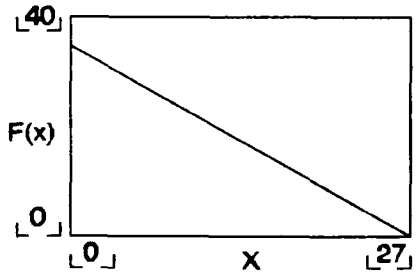
The diagonal line slopes down and to the right so we have a negative slope. The slope is defined as the total vertical distance divided by the total horizontal distance.

$$m = -\frac{HEIGHT}{BASE} \qquad m = 1.333$$

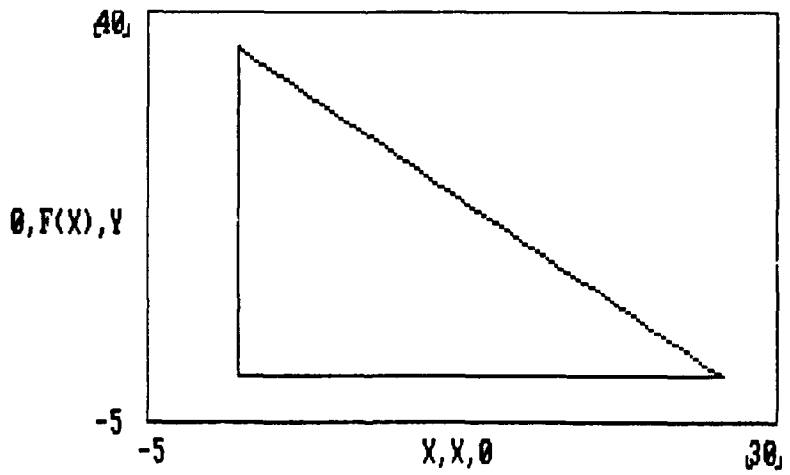
The y intercept is the Y value where X = 0. In this case we have:

b := Height

So the equation of the line is --> F(X) := mX + b



Now we can put all three lines together and form the triangle on a single plot as follows:



Day 2, MathCad Exercise #2, GRADES.mcd

The grades for an exam are recorded in a file named grades.prn. You must read-in the contents of that file and find the following:

MAX_SCORE MIN_SCORE AVERAGE_SCORE VARIANCE STANDARD_DEV

Plot a histogram showing the grade spread by identifying three standard deviations above and below the mean.

GRADES := READPRN(IN) (reading in the grades from the file grades.prn; here IN is the file variable; grades.prn is the filename)

GRADES =

77.7
78
84.7
83
78.2
80.7
90.1
89.8
86.3
83.5
71.1
83
82
86.9
82.3
85.2
82.9
82.1
91.8
83.9
89.2
78.9
95.4
90.1
82.2
91.6
89.4
75.9
82.8
82.1
86.5
86.1
90.1
84.4
84.1
84.4
71.3
84.5
94.7
74.5

MathCad has many built-in Statistics functions. The matrix is a convenient way to specify the argument of the functions.

MAX_SCORE := max(GRADES) MAX_SCORE = 95.4
MIN_SCORE := min(GRADES MIN_SCORE = 71.1
AVERAGE_SCORE := mean((GRADES) AVERAGE_SCORE = 84
VARIANCE := var(GRADES) VARIANCE = 31.1
STANDARD_DEV := stdev(GRADES) STANDARD_DEV = 5.6

In order to plot the histogram that represents the spread of the grades, we must specify a Bin of values to categorize the grades into. The Bin specifies the upper and lower limits on each category. Since we want three standard deviations above and below the mean we can specify the bin values as follows:

r := 0 ..6 (number of elements in the bin)

Bin_r := (AVERAGE_SCORE - 3 STANDARD_DEV) + r STANDARD_DEV

the above equation specifies a Bin with values from -3 standard deviations to +3 standard deviations around the mean.

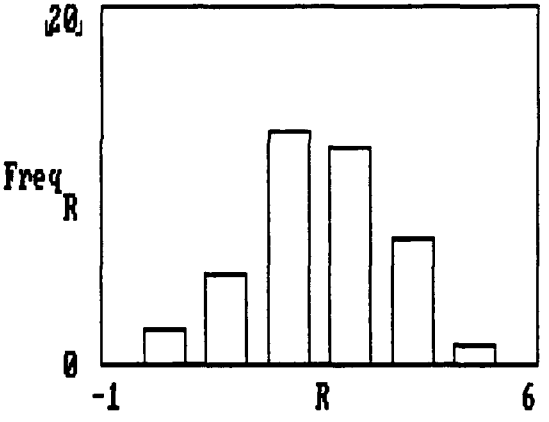
Freq := hist(Bin, GRADES) (the hist function builds the frequency tables using the Bin and the raw data)

$$Bin = \begin{bmatrix} 67.3 \\ 72.9 \\ 78.5 \\ 84 \\ 89.6 \\ 95.2 \\ 100.8 \end{bmatrix} \quad Freq = \begin{bmatrix} 2 \\ 5 \\ 13 \\ 12 \\ 7 \\ 1 \end{bmatrix}$$

R := 0 ..length(Bin) - 2

These range values are used to plot the histogram.

Notice the graph to the right shows the spread of the grades in the input data. You'll notice that the data appears to be "bell shaped" (normal distribution).



The instructor decided to curve the grades according to the following scheme:
The highest grade in the class would be curved up to 100. Each of the other grade would get the same number of points added.
Determine:
Max Score Min Score Average Score Standard Deviation Variance
Then plot the new histogram.

MAX_SCORE = 95.4 (from above)
Delta := 100 MAX_SCORE Delta = 4.6 (curve)
Adj_Grade := GRADES + Delta (this command adds 4.6 points to each grade)

Adj_Grade =

77.7
78
84.7
83
78.2
80.7
90.1
89.8
86.3
83.5
71.1
83
82
86.9
82.3
85.2
82.9
82.1
91.8
83.9
89.2
78.9
95.4
90.1
82.2
91.6
89.4
75.9
82.8
82.1
86.5
86.1
90.1
84.4
84.1
84.4
71.3
84.5
94.7
74.5

GRADES =

82.3
82.6
89.3
87.6
82.8
85.3
94.7
94.4
90.9
88.1
75.7
87.6
86.6
91.5
86.9
89.8
87.5
86.7
96.4
88.5
93.8
83.5
100
94.7
86.8
96.2
94
80.5
87.4
86.7
91.1
90.7
94.7
89
88.7
89
75.9
89.1
99.3
79.1

Adj_Max := max(Adj_Grade)
Adj_Max = 100

Adj_Min := min(Adj_Grade)
Adj_Min = 75.5

Adj_Avg := mean(Adj_Grade)
Adj_Avg = 88.6

Ad_StDev := stdev(Ad_Grade)
Ad_StDev = 5.6

Adj_Var := var(Adj_Grade)
Adj_Var = 31.1

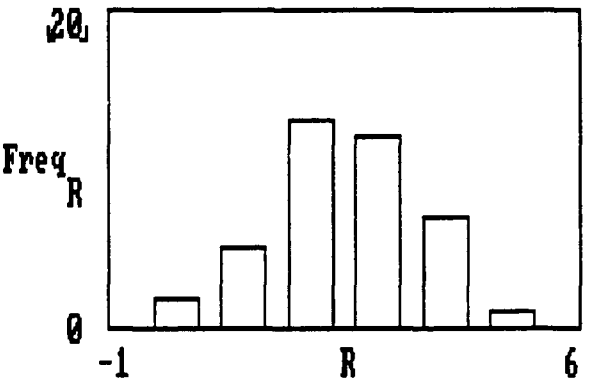
Now we will plot the new data....

We must redefine the Bin since the new values cover a different mean and range.

Bin_r := (Adj_Avg - 3 Adj_Std_Dev) + r Adj_Std_Dev

Freq := hist(Bin,Adj_Grade) R := 0 ..length(Bin) -2

Notice the new plot looks the same as the old plot. This should be obvious since all we did was transform data; we did not change the distribution.



Now we will write the adjusted grades out to disk for future use in a spreadsheet.

WRITEPRN(out) := Adj_Grade

Here out is the file variable and ADJGRADE is the filename. The WRITEPRN command must be in all caps. The WRITEPRN is the complement of the READPRN command used earlier.

Day #2, Mathcad Project

Project #1

You are considering buying a car with a sticker price of $19,000. You have been successful in negotiating a firm sale price of $17,000. You are trading in your old jalopy for $1000. Tags and registration costs are $50 and Ohio state sales tax is 6.5%. You will make a cash down payment of 10% of the total purchase price (after tax and tags and trade-in).

a) Determine the total sale price (after trade-in) for the auto.
b) Determine the amount to be financed.
c) You have the option of one of the following financing arrangements:
   9.9% interest, 54 months
   6.3% interest, 60 months
   11.0% interest, 48 months
d) What is the total cost of payments over the life of the loan in each case above?
e) If you financed the car at the 11% rate as in c) above for the 48 months, and in the 24th month you decide to pay off the loan (no penalty for early pay-off), how much must payment 25 be?
f) Assuming you financed the car for 11% for 48 months (c above), when is the car half paid off? How much are the total payments at this point. How much is the interest?
g) Plot Interest and Principle vs period for the situation in f) above.

Solution:

a) Determine the total sale price (after trade-in) for the auto.

Firm := 17000
Trade_in := 1000
Tags_Reg := 50

The taxable cost is defined as the summation of all the costs listed above minus the trade-in allowance, so:

Taxable_Cost := Firm - Trade_in + Tags_Reg Taxable_Cost = 16050

Total_Cost is no more than the total subtotal cost multiplied by the state tax level of 6.5%. Notice that I have elected to multiply by 1.065 which means I am adding the state tax and getting to the bottom line in one step.

Total_Cost := Taxable_Cost * 1.065 Total_Cost = 17093.25

b) Determine the amount to be financed.

In order to determine the amount financed we must first know the total cost (given above) and the amount of the down payment.

Down_Payment := .1 * Total_Cost Down_Payment = 1709.33

Total financed amount is the difference between total cost and the down payment as shown below:

Finance_amt := Total_Cost - Down_Payment Finance_amt = 15383.93

c) You have the option of one of the following financing arrangements:

9.9% interest, 54 months
6.3% interest, 60 months
11.0% interest, 48 months

Determine the payment under each case (assuming interest rates quoted above are annual interest, compounded monthly; interest rates quoted are not annual effective rate).

$$Pay(Fin, IntRate, Term) := Fin \times \frac{\frac{IntRate}{12}}{1-\left[1+\frac{IntRate}{12}\right]^{-Term}}$$

Determine the payment costs by using a user defined function so that you do not have to re-input the equation each time. Simply issue the parameters for the user defined function and the answer comes out directly as follows:

Pay(Finance_amt,.099,54) = 354.20
Pay(Finance_amt,.063,60) = 299.57
Pay(Finance_amt,.11,48) = 397.61

d) What is the total cost of payments over the life of the loan in each case above?

This is simply the monthly payment multiplied by the term of the loan.

Total1 := Pay(Finance_amt,.099,54) * 54 Total1 = 19126.63
Total2 := Pay(Finance_amt,.063,60) * 60 Total2 = 17973.91
Total3 := Pay(Finance_amt,.11,48) * 48 Total3 = 19085.07

e) If you financed the car at the 11% rate as in c) above for the 48 months, and in the 24th month you decide to pay off the loan (no penalty for early pay-off), how much must payment 25 be?

a := Pay(Finance_amt,.11,48) (for display purposes and ease of entry call the payment function "a")

Simple equation for determining the balance at any time in the loan.

$$Bal(a, IntRate, RemTerm) := a \times \frac{1-\left[1+\frac{IntRate}{12}\right]^{RemTerm}}{\frac{IntRate}{12}}$$

where a= monthly payment IntRate = annual interest rate
RemTerm = remaining term on loan.

Bal(a,.11,24) = 8530.87 (Balance remaining at month 24)

f) Assuming you financed the car for 11% for 48 months (c above), when is the car half paid off? How much are the total payments at this point. How much is the interest?

Use root function to determine the point when half of principle is paid.

x := 20 (guess for the root function)

b := root(Bal(a,.11,x) - .5 * Finance_amt,x) b = 21.39

Suggests there are 21 payments left on the loan after principle is paid)

48 - b = 26.61 (Since there are 48 months in this loan, we must make 27 payments before the loan is half paid off)

$Bal\_rem_{t\_27}$:= Bal(a,.11,21) $Bal\_rem_{t\_27}$ = 7563.85

$Princ_{paid\_27}$:= Finance_amt - $Bal\_rem_{t\_27}$ $Princ_{paid\_27}$ = 7820.08

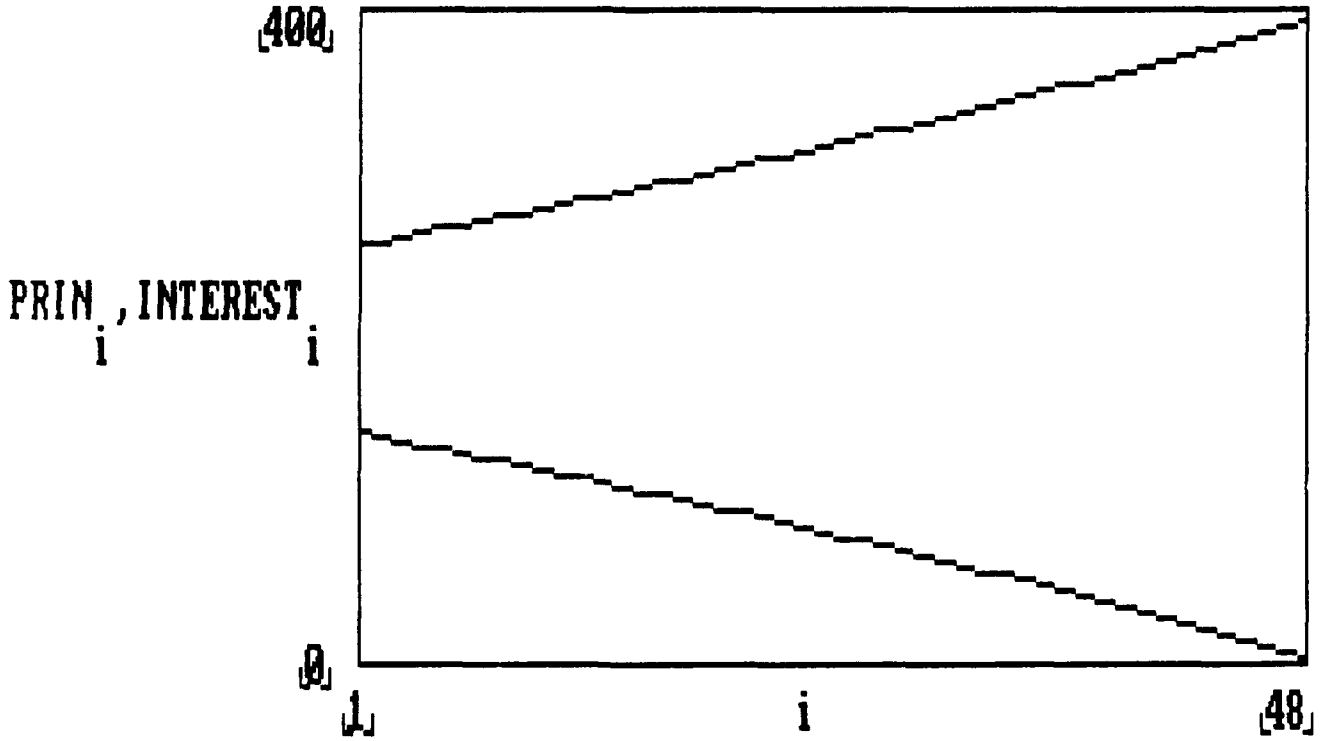
$Interest_{t\_27}$:= Total_Payments - $Princ_{paid\_27}$
$Interest_{t\_27}$ = 2915.27

g) Plot Interest and Principle vs period for the situation in f) above.

1 := 1 ..48
BAL_i := Bal(a,.11,48-i) BAL_0 := Finance_amt
(setting up the balance calculations and initializing the starting balance to the amount financed)

$PRIN_i$:= BAL_{i-1} - BAL_i (Determining the balance at every month)

$INTEREST_i$:= a- $PRIN_i$ (Determining interest payment= delta between monthly payment and principle paid each month)


400
PRIN , INTEREST
i i
0
1
i
48

Day 4, Quattro Exercise #1
LEARNING CURVE EXERCISE

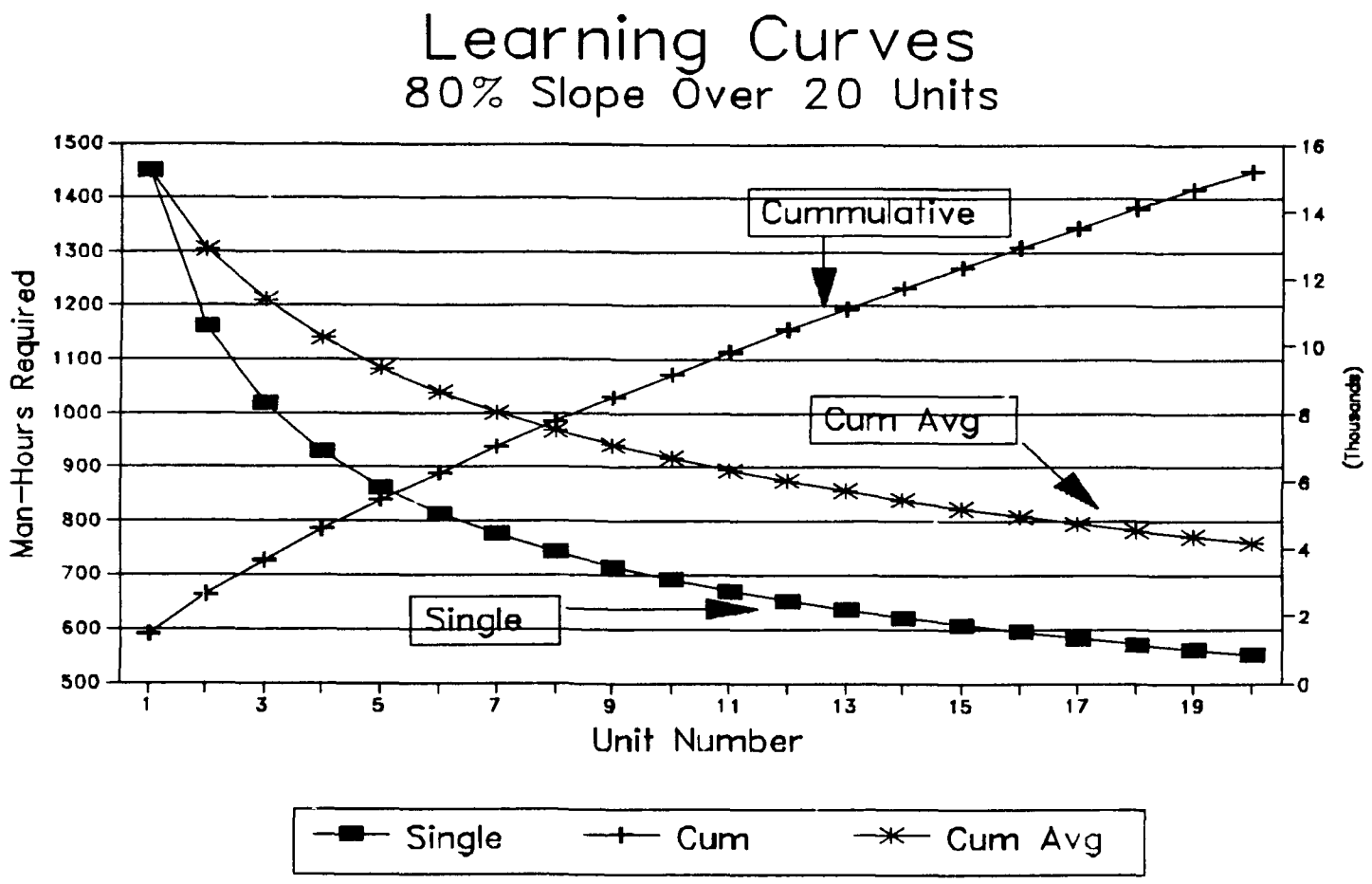
Learning Curve General Formula -> Y = aX^b

where Y is the labor hours for a particular unit
a is the labor hours for the first unit
X is the unit number in the production process

$$b = \frac{\ln(\text{slope of Learning Curve})}{\ln(2)}$$

Here we will assume a= 1450
slope = 80% b= -0.32193
Total X = 20

Unit #	Single Unit Learning Crv	Cumm Learning Crv	Cum Avg Learning Crv
1	1450	1450	1450
2	1160	2610	1305
3	1018	3628	1209
4	928	4556	1139
5	864	5420	1084
6	814	6234	1039
7	775	7009	1001
8	742	7752	969
9	715	8466	941
10	691	9157	916
11	670	9827	893
12	652	10479	873
13	635	11114	855
14	620	11734	838
15	606	12340	823
16	594	12934	808
17	582	13517	795
18	572	14088	783
19	562	14650	771
20	553	15203	760


Learning Curves
80% Slope Over 20 Units
Man-Hours Required
1500
1400
1300
1200
1100
1000
900
800
700
600
500
16
14
12
10
8
6
4
2
0
(Thousands)
1
3
5
7
9
11
13
15
17
19
Unit Number
Cummulative
Cum Avg
Single
Single
Cum
Cum Avg

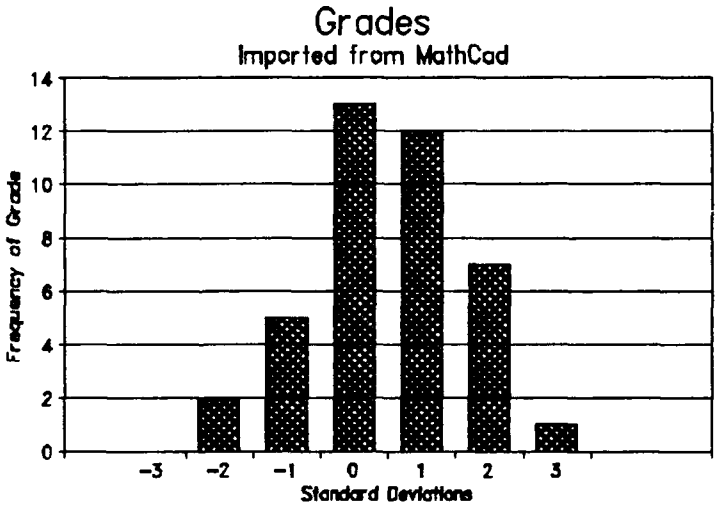
Grade Worksheet

This exercise demonstrates how to import ASCII data from MathCad (or any other package capable of writing ASCII data) into Quattro Pro. The data in the Adjusted Grade column was imported directly from MathCad in number format. This exercise is analygous to the Grade.mcd exercise done in MathCad on Day #2.

Adjusted Grade
82.3
82.6
89.3
87.6
82.8
85.3
94.7
94.4
90.9
88.1
75.7
87.6
86.6
91.5
86.9
89.8
87.5
86.7
96.4
88.5
93.8
83.5
100
94.7
86.8
96.2
94
80.5
87.4
86.7
91.1
90.7
94.7
89
88.7
89
75.9
89.1
99.3
79.1

Stats	
Average	88.635
High	100
Low	75.7
Variance	31.085
Stand Dev	5.575

Std Devs	Bin	Freq
-3	71.9	0
-2	77.5	2
-1	83.1	5
0	88.6	13
1	94.2	12
2	99.8	7
3	105.4	1
		0



Day 5, Project # 2, Spreadsheet Analysis

Westinghouse Corporation builds the AN/APG-68 radar for the F-16 the USAF. They are currently working on a piece of support equipment for the radar that will save the USAF over $100M over the life of the F-16 program.

Westinghouse has already spent $3.0M on RDT&E. They are constructing a new $2.0M specialized production facility, which will be totally obsolete after production of this equipment is complete. Also the facility will have no salvage value after production is complete. The Engineering department predicts that an additional $1.0M will be required for future RDT&E to support this new design. Westinghouse has an agreement with the USAF to spread all fixed costs (past and present) over the entire production lot, allocating a portion of these costs to each unit produced (for the purposes of cash flow calculations, these costs are loaded in month 0 of the production budget).

The company has made detailed predictions of the equipment's production costs. These predictions have proven to be extremely accurate on previous projects of this type and magnitude. Each end item (unit of support equipment) requires materials in the following amounts:

Item	Quantity	Cost/Unit
Widget	2	75000
Thigamajig	5	7000
Thingy	10	5000
Whachamacallit	5	5000
Doodad	10	4000

(the number under the quantity column designates the quantity required per unit of end item)

Direct labor (including benefits) is $40/manhour. The direct labor hours required to assemble the first unit is 3250 manhours. An 80% learning curve, found to be descriptive of labor hours behavior on production runs for past projects, is believed to apply to the current project.

General Operations costs attributable to this project are estimated to be 50000/month. This cost is charged in full for any month where there is production (no matter how low the rate of production).

The pricing department uses the above data to determine costs and pricing for this project. In addition you are to assume that WEC will provide 6600 manhours of effort per month to this project until it is completed. Once the job is completed, all resources are placed on other jobs, such that only the hours required for the task are charged to the project.

All materials required for this item are short lead time. They must be delivered to the production line one month prior to use in the production process. Costs for the materials show up as a charge on the cash flow in the month of delivery. The government pays progress payments to the contractor in the month following delivery of the units. Only whole units are delivered and are billable. Since the fixed costs are to be spread across the production lot, each unit of production will have a fixed cost component.

You are to assume that this contract covers procurement of the support equipment only. Spares (including initial spares) will be purchased under a separate contract. In addition, continuing support after delivery including modifications will be accomplished under a separate contract. Lastly, do NOT apply present value rules to this exercise. This exercise has been simplified for instructional purposes.

Do the following:

1. Prepare a SPREADSHEET or SPREADSHEETS to provide management with the following data:

As a minimum

- Monthly labor available
- Units Produced during each month
- Monthly O&M costs
- Monthly Direct Labor costs
- Monthly Materials costs
- Total Monthly costs
- Monthly Receipts
- Monthly Cash Flow
- Cumulative Cash Flow
- Total Fixed Costs
- Profit (10% of total PRICE) * Note this is NOT the same as 10% of cost!
- Total Price
- Unit Price
- Learning Curve Calculations for this project

2. Plot
   - Total Monthly cost & Total monthly revenue vs month
   - Monthly cash flow vs month
   - Learning Curve - resources required vs unit number
   - Cumulative resources vs units produced

EQUATION FOR LEARNING CURVE: $Y = aX^{b}$

where Y is the manhours required for the Xth unit
a is the manhours spent on the first unit
X is the number of the unit

$$b = \frac{\ln(\textit{slope of learning curve})}{\ln(2)}$$

3. Once 1 and 2 are complete change the learning curve slope to .90 and reaccomplish 1 above again. What is the percent increase/decrease in total price? unit price?

4. Refer back to the 80% slope in 1 above. After production starts, the Air Force decides it does not need 30 of these units and terminates the contract for the convenience of the Government after 19 units are delivered. The termination arrives in time to cancel the materials ordered for all remaining units except those that are currently in the production sequence (work-in-process) The vendors for these materials are gratious enough not to charge a termination fee. Under the Termination for the Convenience clause in the contract, WEC is entitled costs incurred to date and profit on those costs but NOT anticipatory profit or costs. WEC wants $3.2M for the termination. Should the government pay the entire amount? If not, what should the government pay? Why?

NOTE: All numbers should have a whole number FORMAT (this does not mean round the numbers). I will monitor E-MAIL for questions. Any errors or clarifications will be broadcast to the class over E-Mail. All projects are to be turned in with hard copy and disk format. Part 3 should be done on a separate file from parts 1 & 2. This can be very simple if you think about it first. For those using spreadsheets other than QPRO, please save your files in 123 format if possible. MAC users please tell me what package you used so that your spreadsheet may be evaluated.

LEARNING CURVE SPREADSHEET

OBJECTIVE: To calculate the resources required to produce each unit of end item based on the assumption that the production process can be modeled by an 80% learning curve.

3250 First Unit Manhours
0.8 Learning Curve Slope
-0.32193 b (LN.9/LN2)

Y = aX^b Equation of learning curve; Y is the resource requirements for unit X, a is the first unit resource requirements, and X is the unit number.

Units Manufac	Labor Hrs for Unit X	Cum Total Labor Hrs
1	3250.0	3250.0
2	2600.0	5850.0
3	2281.8	8131.8
4	2080.0	10211.8
5	1935.8	12147.7
6	1825.5	13973.1
7	1737.1	15710.2
8	1664.0	17374.2
9	1602.1	18976.3
10	1548.7	20525.0
11	1501.9	22026.8
12	1460.4	23487.2
13	1423.2	24910.4
14	1389.7	26300.1
15	1359.1	27659.2
16	1331.2	28990.4
17	1305.5	30295.9
18	1281.7	31577.6
19	1259.6	32837.1
20	1238.9	34076.1
21	1219.6	35295.7
22	1201.5	36497.2
23	1184.4	37681.6
24	1168.3	38849.9
25	1153.0	40002.9
26	1138.6	41141.5
27	1124.8	42266.3
28	1111.7	43378.1
29	1099.2	44477.3
30	1087.3	45564.7

The bin below is used to determine the number of ful units completed each month. The FREQUENCY function is used to do this calculation The bin values are determin by the cumulative hours available each month.

Bin	Units Produced
6600.0	2
13200.0	3
19800.0	4
26400.0	5
33000.0	5
39600.0	5
46200.0	6
52800.0	0
59400.0	0
66000.0	0
72600.0	0
	0

Fixed Costs

RDT&E (sunk)	3000000
RDT&E future	1000000
Prod'n Facil	2000000

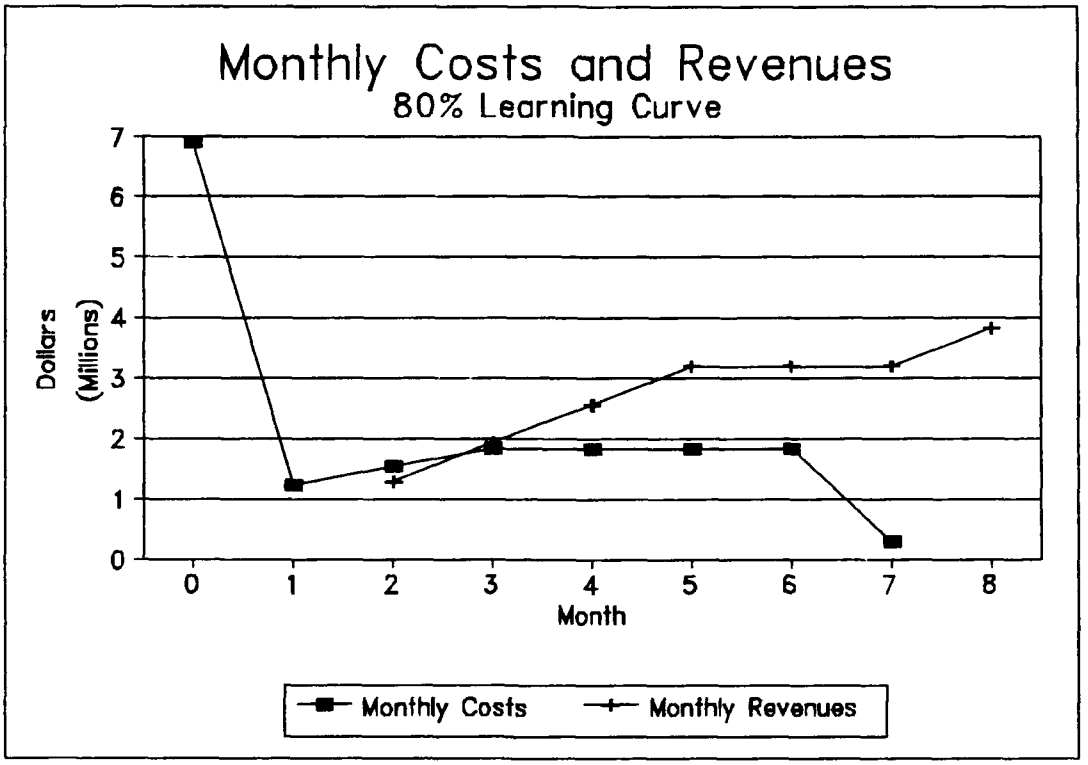
Labor Rate (per mhr) 40

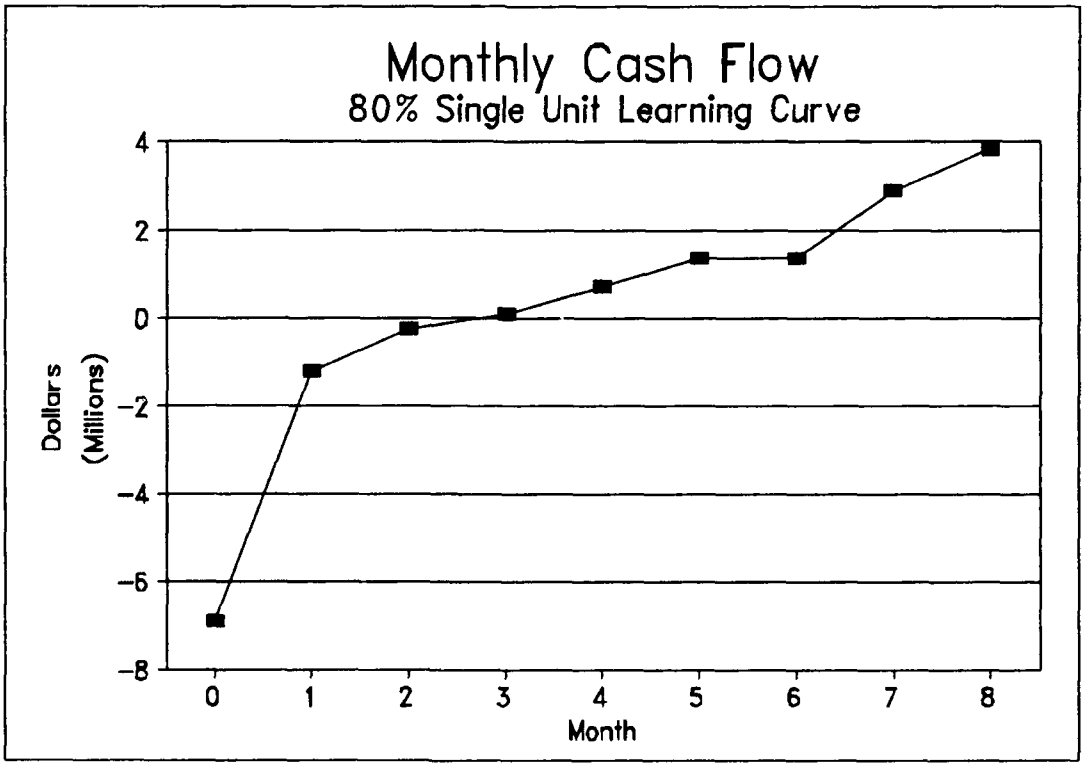
Materials

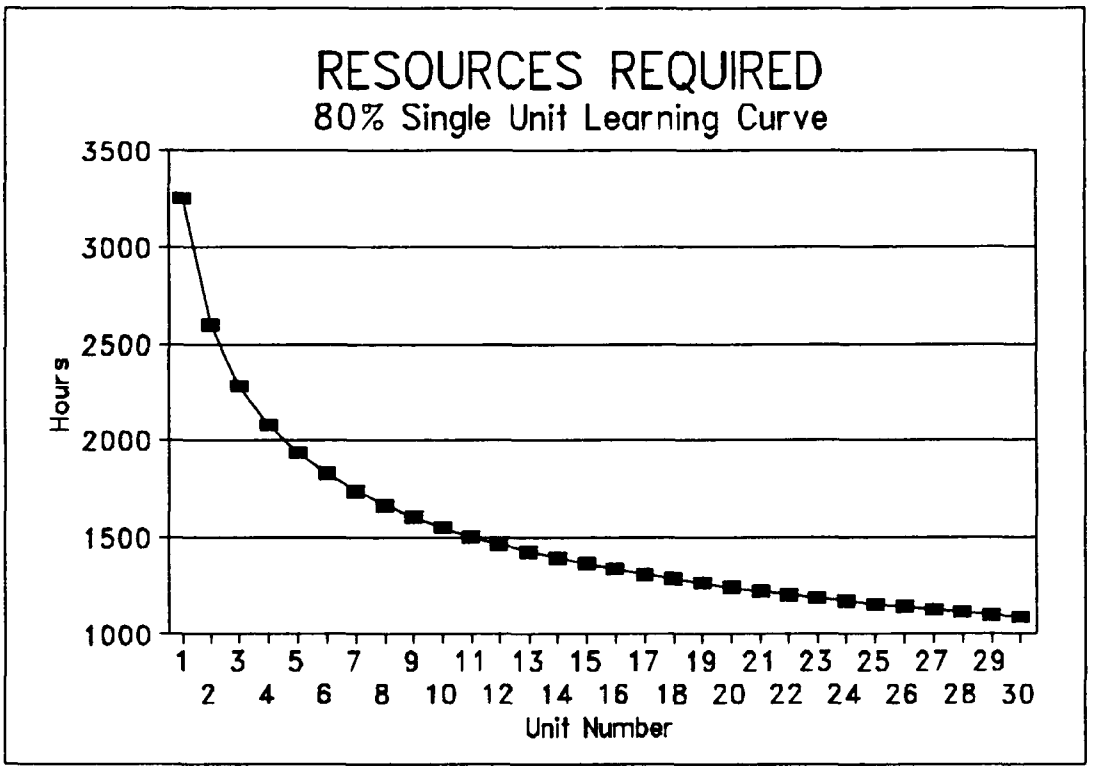
Item	Quantity	Unit $	Tot Cost
Widget	2	75000	150000
Thigamajig	5	7000	35000
Thingy	10	5000	50000
Whachamacallit	5	5000	25000
Dooded	10	4000	40000
		Material Cost/Un	300000

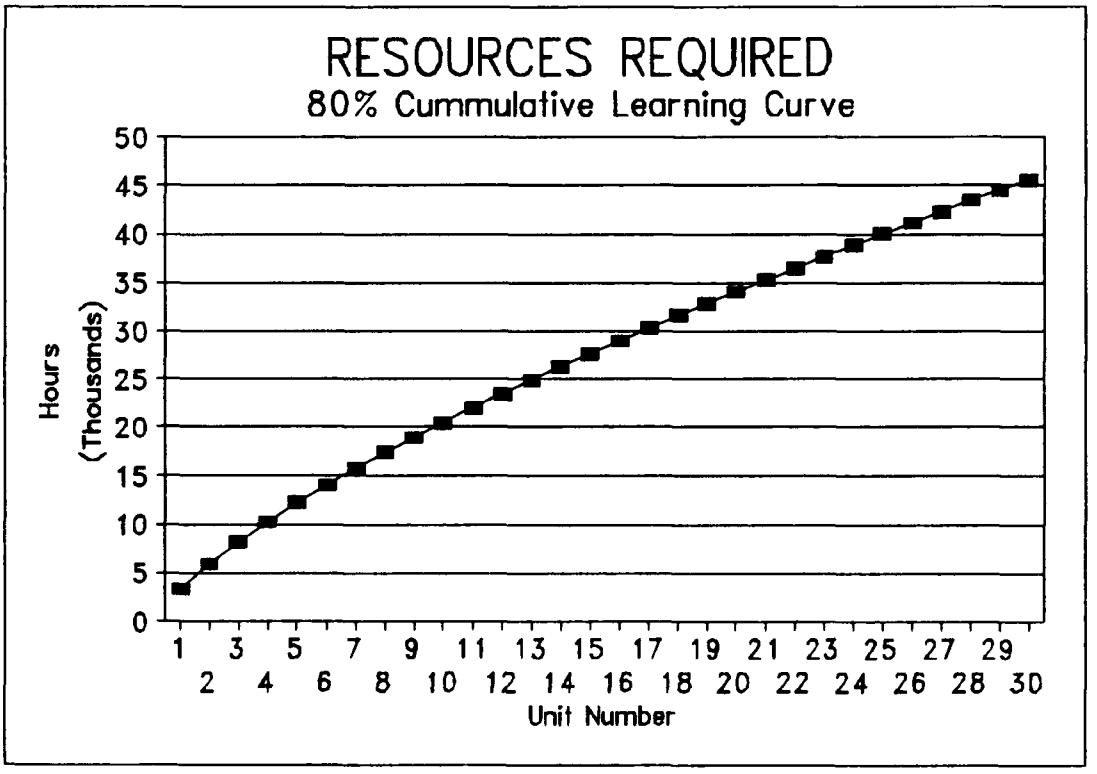
Labor Hrs Avail/mth	6600
O&M cost/mth	50000

Month	Labor Hrs Avail	Units Prod	Monthly O&M	Direct Labor	Matrls	Fixed Costs	Monthly Tot Costs	Monthly Receipts	Mthly Cas Flow	Cum Cash Flow
0					900000	6000000	6900000		-6900000	-6900000
1	6600	2	50000	264000	900000		1214000		-1214000	-8114000
2	6600	3	50000	264000	1200000		1514000	1272043	-241957	-8355957
3	6600	4	50000	264000	1500000		1814000	1908065	94065	-8261891
4	6600	5	50000	264000	1500000		1814000	2544087	730087	-7531805
5	6600	5	50000	264000	1500000		1814000	3180109	1366109	-6165696
6	6600	5	50000	264000	1500000		1814000	3180109	1366109	-4799588
7	5965	6	50000	238586			288586	3180109	2891522	-1908065
8								3816130	3816130	1908065
	Tot Units	30			Total Costs		17172586	19080651		
					Profit		1908065			
					Total Price		19080651			
					Unit Price		636022			


Monthly Costs and Revenues
80% Learning Curve
Dollars (Millions)
7
6
5
4
3
2
1
0
0 1 2 3 4 5 6 7 8
Month
Monthly Costs
Monthly Revenues


Monthly Cash Flow
80% Single Unit Learning Curve
Dollars (Millions)
4
2
0
-2
-4
-6
-8
0 1 2 3 4 5 6 7 8
Month


RESOURCES REQUIRED
80% Single Unit Learning Curve
Hours
3500
3000
2500
2000
1500
1000
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Unit Number


RESOURCES REQUIRED
80% Cummulative Learning Curve
Hours
(Thousands)
50
45
40
35
30
25
20
15
10
5
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Unit Number

LEARNING CURVE SPREADSHEET

OBJECTIVE: To calculate the resources required to produce each unit of end item based on the assumption that the production process can be modeled by an 90% learning curve.

3250 First Unit Manhours
0.9 Learning Curve Slope
-0.152 b (LN.9/LN2)

Y = aX^b Equation of learning curve; Y is the resource requirements for unit X, a is the first unit resource requirements, and X is the unit number.

Units Manufac	Labor Hrs for Unit X	Cum Total Labor Hrs
1	3250.0	3250.0
2	2925.0	6175.0
3	2750.2	8925.2
4	2632.5	11557.7
5	2544.7	14102.4
6	2475.2	16577.5
7	2417.8	18995.4
8	2369.3	21364.6
9	2327.2	23691.8
10	2290.2	25982.1
11	2257.3	28239.4
12	2227.6	30467.0
13	2200.7	32667.7
14	2176.0	34843.7
15	2153.3	36997.1
16	2132.3	39129.4
17	2112.8	41242.2
18	2094.5	43336.7
19	2077.3	45414.0
20	2061.2	47475.2
21	2046.0	49521.2
22	2031.6	51552.8
23	2017.9	53570.7
24	2004.9	55575.5
25	1992.5	57568.0
26	1980.6	59548.6
27	1969.3	61517.9
28	1958.4	63476.4
29	1948.0	65424.4
30	1938.0	67362.4

The bin below is used to determine the number of ful units completed each month. The FREQUENCY function is used to do this calculation The bin values are determin by the cumulative hours available each month.

Bin	Units Produced
6600.0	2
13200.0	2
19800.0	3
26400.0	3
33000.0	3
39600.0	3
46200.0	3
52800.0	3
59400.0	3
66000.0	4
72600.0	1
	0

Fixed Costs	
RDT&E (sunk)	3000000
RDT&E future	1000000
Prod'n Facil	2000000

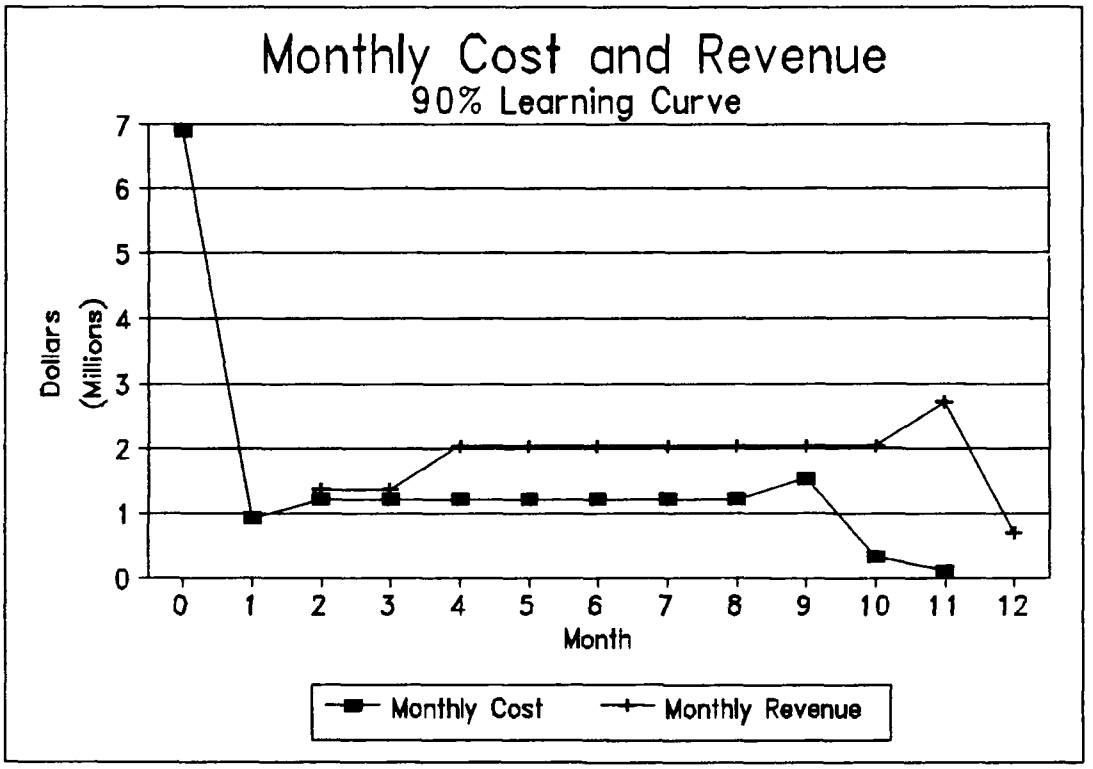
Labor Rate(per mhr)	40

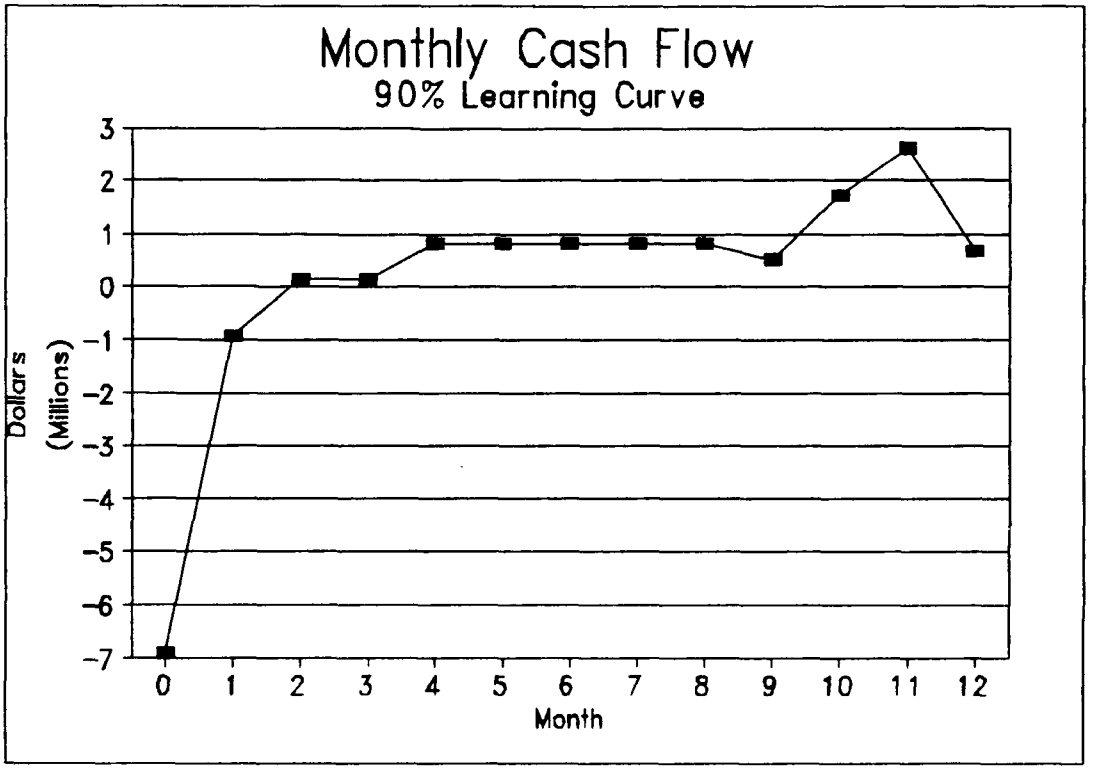
Materials Item	Quantity	Unit Cost	Tot Cost
Widget	2	75000	150000
Thigamaji	5	7000	35000
Thingy	10	5000	50000
Whachamac	5	5000	25000
Doodad	10	4000	40000
			300000
Material Cost/Unit			300000

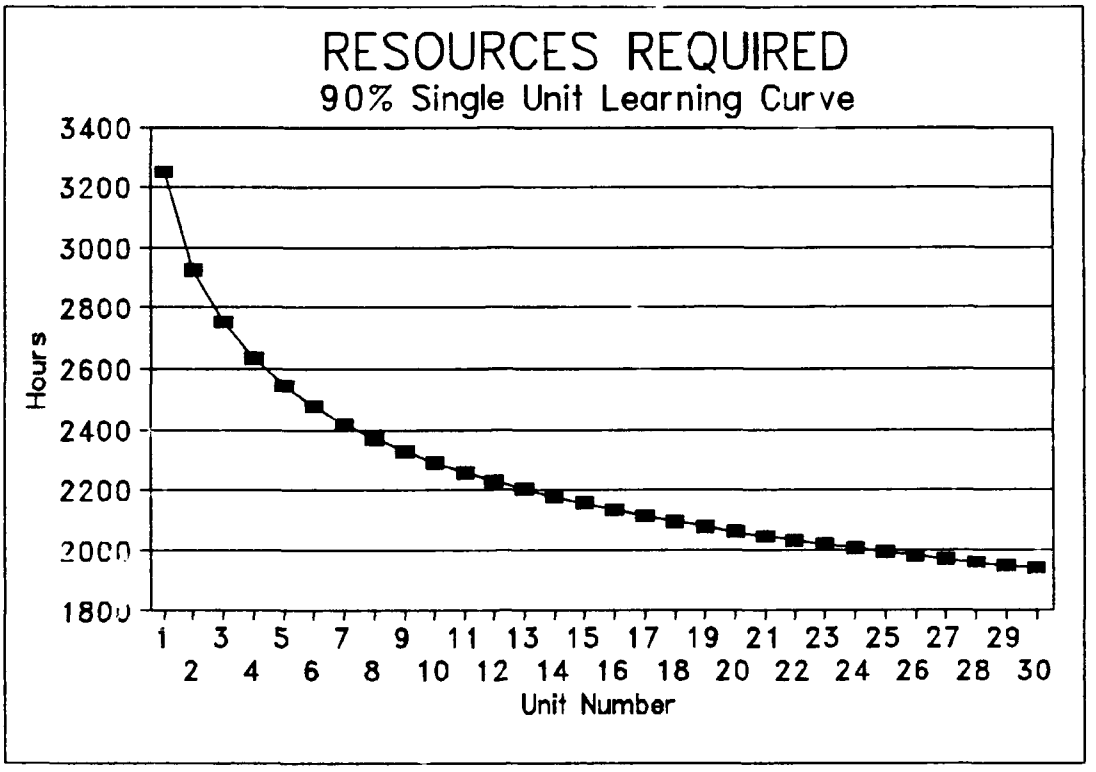
Labor Hrs Avail/mth	6600
O&M cost/mth	50000

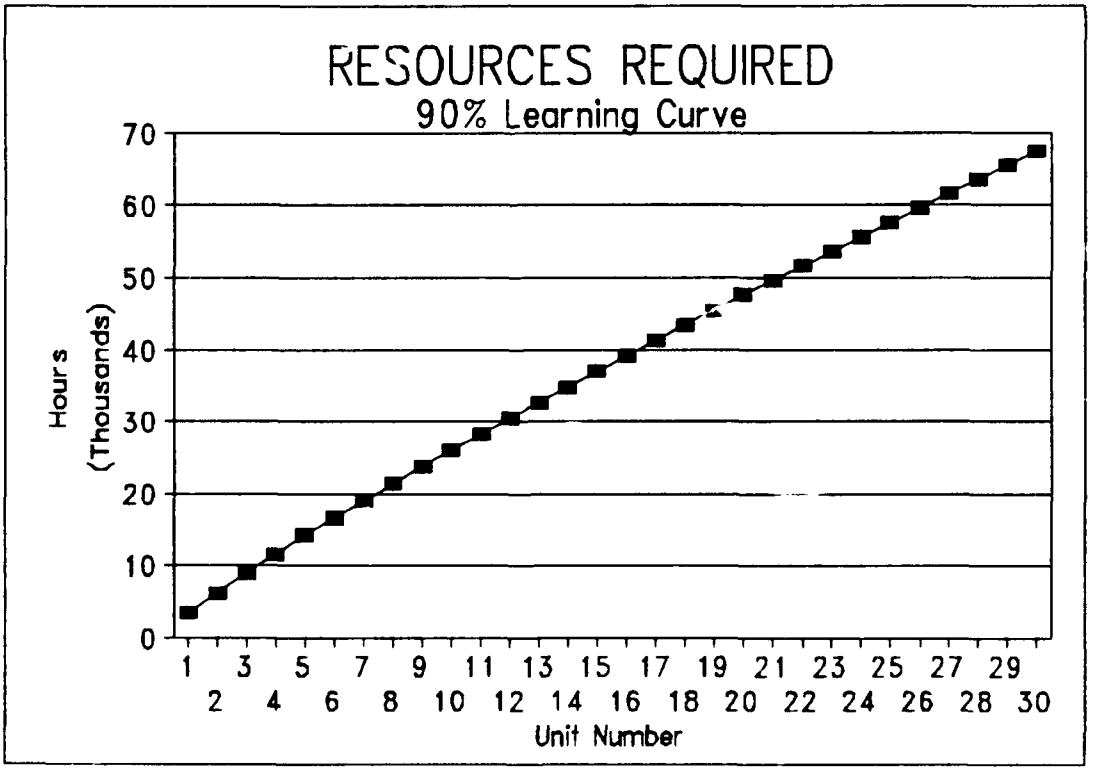
Month	Labor Hrs Avail	Units Prod	Monthly O&M	Direct Labor	Matrls	Fixed Costs	Monthly Tot Costs	Monthly Receipts	Mthly CasCum Cash Flow	Cum Cash Flow
0					900000	6000000	6900000		-6900000	-6900000
1	6600	2	50000	264000	600000		914000		-914000	-7814000
2	6600	2	50000	264000	900000		1214000	1351446	137446	-7676554
3	6600	3	50000	264000	900000		1214000	1351446	137446	-7539108
4	6600	3	50000	264000	900000		1214000	2027169	813169	-6725939
5	6600	3	50000	264000	900000		1214000	2027169	813169	-5912770
6	6600	3	50000	264000	900000		1214000	2027169	813169	-5099601
7	6600	3	50000	264000	900000		1214000	2027169	813169	-4286433
8	6600	3	50000	264000	900000		1214000	2027169	813169	-3473264
9	6600	3	50000	264000	1200000		1514000	2027169	513169	-2960095
10	6600	4	50000	264000			314000	2027169	1713169	-1246926
11	1363	1	50000	54520			104520	2702892	2598372	1351446
12								675723	675723	2027169
Tot Units		30								

Total Costs	18244520	20271689	
Profit	2027169		
Total Price	20271689		
Unit Price	675723		
Unit Price(80%LC)	636022	Total Price(80%LC)	19080667
% inc unit price	6.2%	% inc total price	6.2%


Monthly Cost and Revenue
90% Learning Curve
Dollars (Millions)
7
6
5
4
3
2
1
0
0 1 2 3 4 5 6 7 8 9 10 11 12
Month
Monthly Cost
Monthly Revenue


Monthly Cash Flow
90% Learning Curve
Dollars (Millions)
3
2
1
0
-1
-2
-3
-4
-5
-6
-7
0 1 2 3 4 5 6 7 8 9 10 11 12
Month


RESOURCES REQUIRED
90% Single Unit Learning Curve
Hours
3400
3200
3000
2800
2600
2400
2200
2000
1800
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
Unit Number


RESOURCES REQUIRED
90% Learning Curve
Hours
(Thousands)
70
60
50
40
30
20
10
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
Unit Number

Fixed Costs	
RDT&E (sunk)	3000000
RDT&E future	1000000
Prod'n Facil	2000000

Labor Rate(per mhr)	40

Materials		Unit	
Item	Quantity	Cost	Tot Cost
Widget	2	75000	150000
Thigamaji	5	7000	35000
Thingy	10	5000	50000
Whachamac	5	5000	25000
Doodad	10	4000	40000
		Material Cost/Unit	300000

Labor Hrs Avail/mth	6600
O&M cost/mth	50000

Month	Labor Hrs Avail	Units Prod	Monthly O&M	Direct Labor	Matrls	Fixed Costs	Monthly Tot Costs	Monthly Receipts	Mthly Cas Flow	Cum Cash Flow
0					900000	6000000	6900000		-6900000	-6900000
1	6600	2	50000	264000	900000		1214000		-1214000	-8114000
2	6600	3	50000	264000	1200000		1514000	1272044	-241956	-8355956
3	6600	4	50000	264000	1500000		1814000	1908066	94066	-8261890
4	6600	5	50000	264000	1200000		1514000	2544088	1030088	-7231802
5	6438	5	50000	257520			307520	3180110	2872590	-4359212
6								3180110	3180110	-1179102
7										-1179102
8										-1179102
	Tot Units	19			Total Costs		13263520	12084418		
					Profit		1473724			
					Total Price		14737244			
					Unit Price		775644			
					Unit Price Paid		636022			

Termination Liability

Government liable for unpaid fixed costs + profit on those costs

Based on a production lot of 30 units, the per unit cost was $636022.
Since the government terminated for its convenience after only 19 units
the new per unit cost is $775,644 as shown below.
The government is liable for the difference between the per unit
costs of $636,022 and $775,644.

Per Unit cost (30 units)	636022
Per unit cost (19 units)	775644
Units delivered	19
Total already paid	12084418
Total Billable	14737244
Total owed WEC	2652826

Government should settle for $2,652,826. firm

Day #6, WordPerfect Exercise

1. Given the following excerpts from the United States Constitution, type in the first paragraph.

2. Merge the file CON.TXT from the disk provided at the end of the first paragraph.

3. Edit the document to make it look like the excerpt given. Remember to spell check the document before completion.

CON.TXT

CONSTITUTION OF THE UNITED STATES

We the People of the United States, in Order to form a more perfect Union, establish Justice, insure domestic Tranquility, provide for the common defence, promote the general Welfare, and secure the Blessings of Liberty to ourselves and our Posterity, do ordain and establish this Constitution for the United States of America.

ARTICLE I.

SECTION 1. All legislative Powers herein granted shall be vested in as Congress of the United States, which shall consist of a Senate and House of Representatives.

SECTION 2. [1]The House of Representatives shall be composed of Members chosen every second Year by the People of the several States and the Electors in each State shall have the Qualifications requisite for Electors of the most numerous Branch of the state legislature.

[2]No person shall be a Representative who shall not have attained to the Age of twenty five Years, and been seven Years a Citizen of the United States, and who shall not, when elected, be an Inhabitant of that State in which he shall be chosen.

[3]Representatives and direct Taxes shall be apportioned among the several States which may be included within this Union, according to their respective Numbers, which shall be determined by adding to the whole Number of free Persons, including those bound to Service for a Term of Years, and excluding Indians not taxed, three fifths of all other Persons. The actual Enumeration shall be made within three Years after the first meeting of the Congress of the United States, and within every subsequent Term of ten years, in such Manner as they shall by Law direct. The number of Representatives shall not exceed one for every thirty thousand, but each state shall have at Least on Representative; and until such enumeration shall be made, the State of New Hampshire

shall be entitled to chose three, Massachusetts eight, Rhode-Island and Providence Plantations one, Connecticut five, New York six, New Jersey four, Pennsylvania eight, Delaware one, Maryland six, Virginia ten, North Carolina five, South Carolina five, and Georgia three.

[4]When vacancies happen in the Representation from any State, the Executive Authority thereof shall issue Writs of Election to fill such Vacancies. The House of Representatives shall chose their Speaker and other Officers; and shall have the sole Power of Impeachment.

SECTION 3. [1]The Senates of the United States shall be composed of two Senators from each State, chosen by the Legislature thereof, for six Years; and each Senator shall have one Vote.

[4]Neither House, during a Session of Congress, shall without the Consent of the other, adjourn for more than three days, nor to any other Place than that in which the two Houses shall be sitting.

SECTION 6. [1]The Senators and Representatives shall receive a Compensation for their Services, to be ascertained by Law, and paid out of the Treasury of the United States. They shall in all Cases, except Treason, Felony, and Breach of the Peace, be privileged from Arrest during their Attendance at the Session of their respective Houses, and in going to and returning from the same; and for any Speech or Debate in either House, they shall not be questioned in any other Place.

[2]No Senator or Representative shall, during the Time for which he was elected, be appointed to any civil Office under the Authority of the United States, which shall have been created, or the Emoluments whereof shall have been encreased during such time: and no Person holding any Office under the United States, shall be a Member of either House during his Continuance in Office.

Day #7, WordPerfect Exercise - Import Graphics and ASCII data into WordPerfect.

1. Retrieve the file ASCII file named BUDGET.DOC from the disk provided into WordPerfect.

2. Import the graphic file named BUDGET.PIC (a Quattro Pro graphic) into the same document, placing the graphic between the first and second paragraph of the text.

3. The resultant document should be formatted as follows:

a. The graphic must comply with the following:
- be at least 1/2 page in size
- be centered on the page between the right and left margins
- no text may be wrapped around the side of the graphic
- the graphic caption should be centered below the graphic outside the graphic definition box.

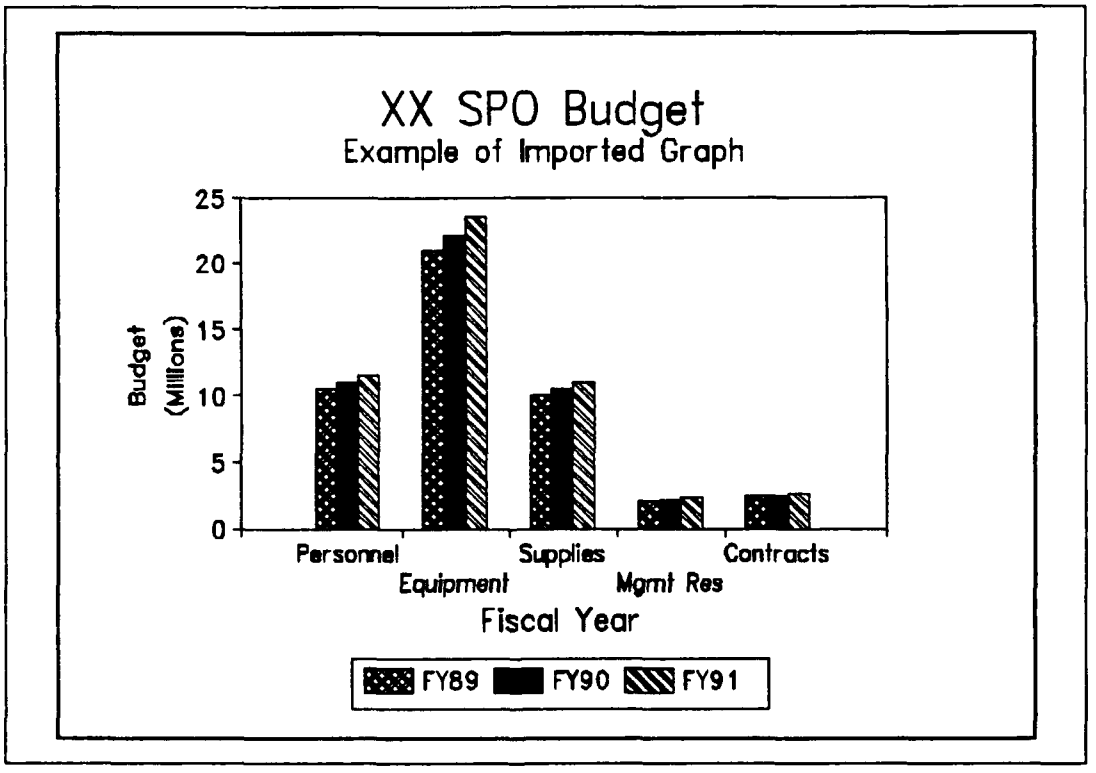
b. The text should comply with the following:
- left and right margins be 1.25"
- top margin be 1.0"
- bottom margin be 0.75"
- page number be centered on the bottom of the page
- double space all text
- left justify all text

BUDGET.DOC

The purpose of this exercise is to demonstrate the capabilities of WordPerfect to integrate text and graphics into a simple document. WordPerfect has the ability to import graphic formats including the popular .PIC (Lotus Picture) file format used by many popular spreadsheets.

Notice the ease of importing graphics into documents. Graphics tend to spice up a document and make it more interesting for the reader to read. With WordPerfect the user can do many of the conventional desktop publishing tasks that just a few years ago required a separate software package to accomplish.

BUDGET.PIC



Solution:

The purpose of this exercise is to demonstrate the capabilities of WordPerfect to integrate text and graphics into a simple document. WordPerfect has the ability to import graphic formats including the popular .PIC (Lotus Picture) file format used by many popular spreadsheets.

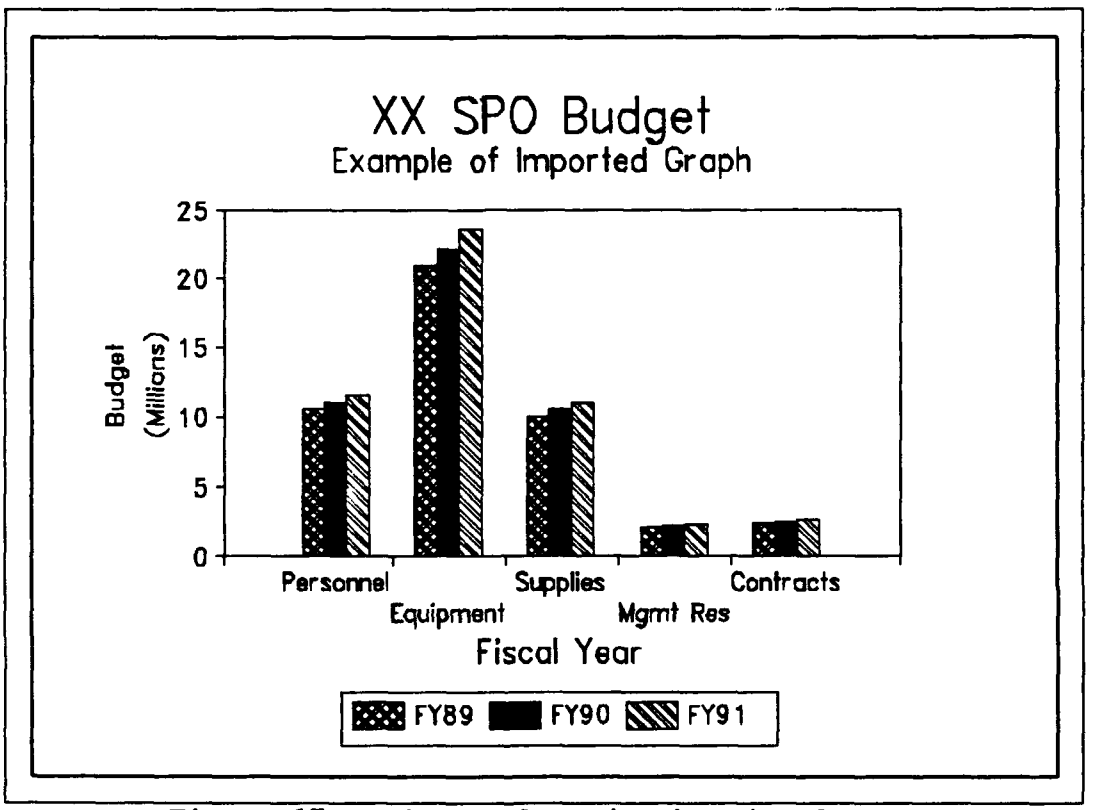


Figure 17: Budget Allocation by Fiscal Year

Notice the ease of importing graphics into documents. Graphics tend to spice up a document and make it more interesting for the reader to read. With WordPerfect the user can do many of the conventional desktop publishing tasks that just a few years ago required a separate software package to accomplish.

QMGT 290 PROJECT #3
Word Processing

Refer back to Project #2, Spreadsheet Analysis. Under part 4 you were to assume cancellation of the contract at 19 units vice the 30 originally contracted for. You were asked to determine if the $3.2M that WEC wants for cancellation of the contract is reasonable. You just completed your analysis and are about to make your recommendations to your boss. Your boss asked you to document the procurement files to reflect your recommendation.

You must include the following in your report:

1. Complete explanation of your rationale for your recommendation to include relevant parts of your spreadsheet used to make your decision.

2. At least two graphs (imported into your word processor) showing relevant data to support your conclusions.

3. A complete explanation of each of the above graphs and how they support your conclusions.

4. Any additional information you think is necessary to document your recommendation.

Guidelines: Your report must demonstrate that you can:

Center Text
Bold Text
Underline Text
Use Large Font
Indent Text
Set Left & Right Margins to be 1.25"
Set Top and Bottom Margins to be 1.00"
Double space text (graphs and spreadsheet tables should be single spaced)
Assure NO widow or orphan lines or words
Set page numbering so that all pages except the cover page are numbered; numbers should appear centered at the bottom of the page

The cover page should include the following:

- Course Title
- Assignment Name
- Names of Group Members
- Date

Graphs must have boxes clearly marked around them (automatically produced in Word Perfect) and with right justified titles (e.g. Figure 1: Learning Curve). All graphs must be approximately 1/2 page in size and centered between the left and right margins. Figures should not appear in the document until introduced in text.

TEXT MUST NOT WRAP AROUND YOUR GRAPHS OR TABLES!!!

Solution:

TERMINATION CASE

Westinghouse Corporation is providing a new piece of support equipment to support the AN/APG-68 radar for the F-16. In the original USAF buy, the government contracted for **30** end items. Due to changes in requirements, the USAF has determined that **19** units will be sufficient to meet the need. As such, the government took action to terminate the contract for the convenience of the government.

The USAF is now faced with the termination liability issue for the contract. Under the contract the contractor is entitled to costs incurred plus profit on those costs incurred. The contractor is NOT entitled to anticipatory profit or costs. WEC was able to terminate its supplier contracts with no penalty. WEC's prodution plan is displayed below:

Month	Units Prod	Monthly O&M	Direct Labor	Matrls	Fixed Costs	Monthly Tot Costs	Monthly Receipts	Mthly Cas Flow	Cum Cash Flow
0				900000	6000000	6900000		-6900000	-6900000
1	2	50000	264000	900000		1214000		-1214000	-8114000
2	3	50000	264000	1200000		1514000	1272044	-241956	-8355956
3	4	50000	264000	1500000		1814000	1908066	94066	-8261890
4	5	50000	264000	1200000		1514000	2544088	1030088	-7231802
5	5	50000	257520			307520	3180110	2872590	-4359212
6							3180110	3180110	-1179102
Total Units	19		Total Costs			13263520	12084418		
			Profit			1473724			
			Total Price			14737244			
			Unit Price (paid)			636022			
			Unit Price (billable)			775644			

As the table shows, the billable cost per unit of end item was determined to be $636,022 per unit with a lot size of 30 units. Adjusting for the reduced lot size (19) the billable cost is $775,644. As such the government is liable for $139,622 more per unit. This

increase in unit size is due primarily to the unpaid non-recurring costs associated with development of the system.

WEC has requested $3.2M for termination but this office finds that government liability is limited to $2,652,826 as shown below.

Per Unit Cost (30 units):	$636,022
Per Unit Cost (19 units):	$775,644
Units Delivered:	19
Total Already Paid:	$12,084,418
Total Billable:	$14,737,244
Total owed WEC:	$2,652,826

Note: The attached figures support this analysis.

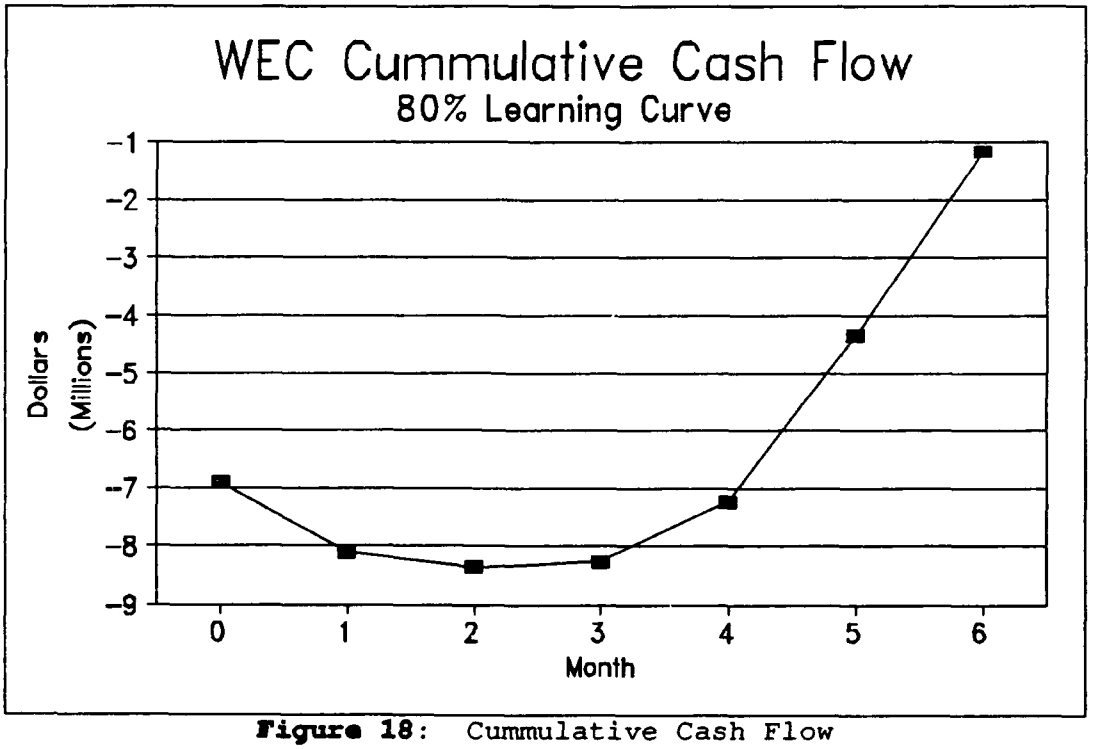


Figure 18: Cummulative Cash Flow

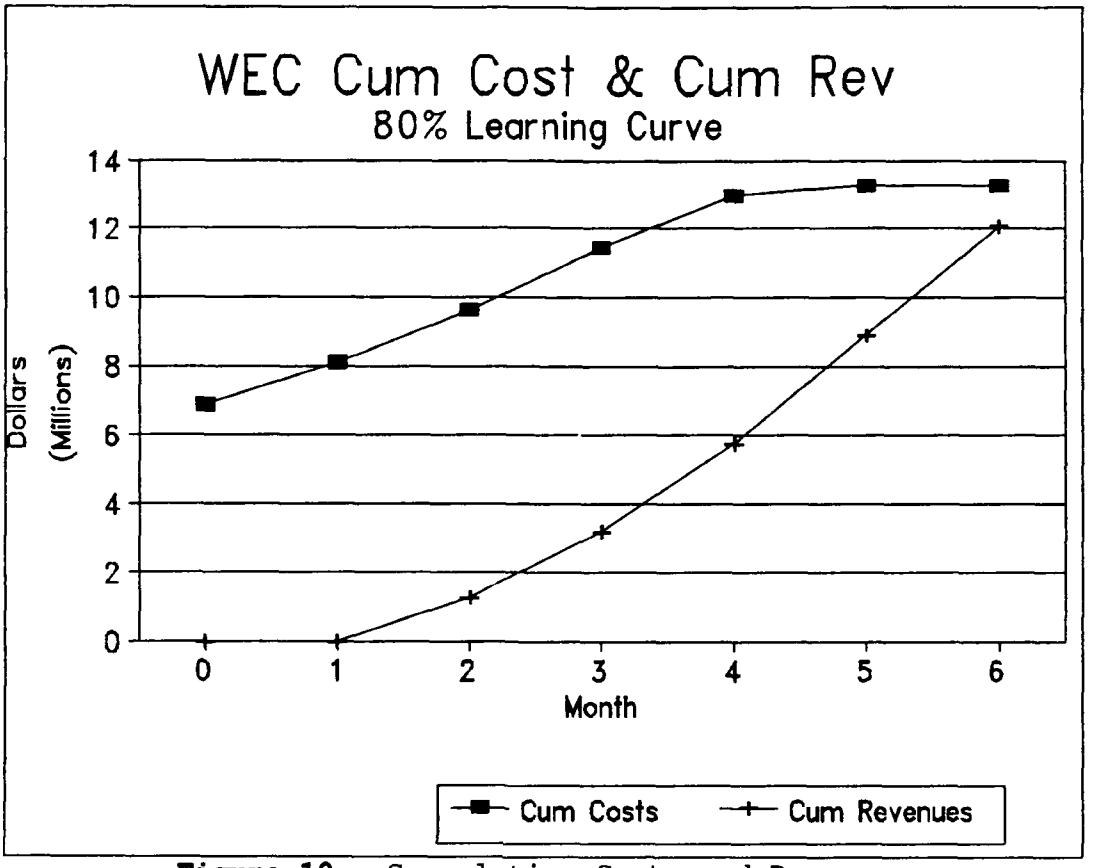


Figure 19: Cummulative Costs and Revenues

Bibliography

1. Air Force Institute of Technology. 1987-1989 Catalog. Wright-Patterson AFB OH, AFIT/RR, 1988.

2. Bass, Brad "Government Micro Users Report Problems," Government Computer News, 5:61-62 (15 August 1986)

3. Brownell, Greg. Computers and Teaching. New York: West Publishing Company, 1987 ISBN 0-314-28487-7

4. Coulson, John E. Responsiveness as an Organizing Principle in the Analysis and Design of Education Systems. Technical Document. System Development Corporation, Santa Monica, 22 August 1963.

5. Flygare, Thomas J. "Cyberphobia Redux," Phi Delta Kappan, 69:528-529 (March 1988)

6. Grant, Phillip C. "Making the Most of a Training Session, ManagementSolution, 32:39-45 (June 1987).

7. Hansgate, John P. "Avoiding Overload," Data Training, 5:17-18, (June 1986).

8. Harrison, Gay L. A Determination of Perceived Computer Literacy and Computer Training Needs of Air Force Institute of Technology Graduate Class 89S/D. MS Thesis AFIT/GSM/LSQ/89S-16. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1989.

9. James, Major Thomas G. Jr. ACSC Introduction To Microcomputers For Air Force Leaders, ACSC Paper 87-1330. Air Command and Staff College, Maxwell AFB AL, April 1987.

10. Kelleher, Joanne. "Step by Step Training," Personal Computing, 10:111-117 (December 1986)

11. Lenz, Richard C. An Analysis of Computer Skills Possessed by AFIT Class 89S/D. MS Thesis AFIT/GLM/LSQ/88S-41. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1988.

12. Lineback, L. Kent. "Corporate Classroom," Executive Female, 36+ (September-October 1987).

13. McPartlin, John P. "The Trouble With PC Training," Information Week, 52 (24 October 1984).

14. Needle, David. "Selective Shopping Yields Greatest Training Payback," Computerworld, XX:57+ (5 May 1986).

15. Nelson, R. Ryan and Paul Cheney. "Training Today's User," Datamation, 33:121-122 (15 March 1987)

16. Novak, Joseph D. and D. Bob Gowin. Learning How to Learn. New York. Cambridge University Press, 1984 ISBN 0-521-31926-9.

17. Orgel, Robert F. and Ann I. Kelley. "It's More Than Hocus Pocus," Data Training, 5:21-22,60 (July 1986).

18. "Payback of PC Training," Personal Computing, 12:146-152 (November 1988).

19. Peschke, Lieutenant Colonel Richard E., Associate Professor of Quantitative Management. Personal Interview. Air Force Institute of Technology, School of Systems and Logistics, Wright-Patterson AFB OH, September 1989.

20. Simonson, Michael, et al. "Development of a Standardized Test of Computer Literacy and Computer Anxiety Index," Journal of Education Computing Research, 3:232-240 (No 2 1987).

21. Spain, T. "How State Governments Are Helping To Define Computer Literacy," Electronic Learning, 5:42-44 (Spring 1985).

22. Suydam, Bill. "Military Micros," Government Computer News, 7:63-65 (18 March 1986)

23. Troutner, Joanne. "Toward a Definition of Computer Literacy," Journal of Computers in Mathematics and Science Teaching, 5:12-13 (Spring 1985).

24. Ussery, Jean and Bill Marriott. "What To Look For In End-User Training," Data Training, 5:20-21,80 (November 1986).

25. Witten, Charles S. D. Development of IS2100 An Information Systems Laboratory, MS Thesis. Naval Post Graduate School, Monterey CA, March 1985 (AD-A156 848).

26. Zemke, Ron and Susan. "30 Things We Know for Sure About Adult Learning," Training, 57-61 (July 1988).

Vita

Captain Stephen M. Heaps was born on 6 March 1961 at Geisinger Medical Center, Danville, Pennsylvania. He was a Rotary International Youth Exchange student to the Republic of Indonesia during his senior year of high school. He graduated from Bloomsburg Senior High School in Bloomsburg, Pennsylvania in May 1979. He received a four year Reserve Officer Training Corp (ROTC) scholarship to attend Embry-Riddle Aeronautical University starting in September 1979. Steve graduated from Embry-Riddle with a Bachelor of Science in Aeronautical Engineering in April 1983 and was commissioned a Second Lieutenant in the United States Air Force. Steve was assigned to the 4950th Test Wing Aircraft Modification Directorate at Wright-Patterson Air Force Base, Ohio in January 1984 where he served as a modifications operations officer and a modifications project officer. In February 1986, he was transferred to the F-16 System Program Office, Aeronautical Systems Division, Wright-Patterson Air Force Base, Ohio. He served as the program manager for The Royal Netherlands Air Force F-16 program, Royal Norwegian Air Force F-16 Program, Republic of Singapore Air Force F-16 program, and project officer for integration efforts on the Belgian Air Force and Royal Danish Air Force F-16 programs. Steve attended Squadron Officers School from March - May 1988. He entered the School of Systems and Logistics, Air Force Institute of Technology, in May 1989. Steve is married to the former Lorrie Ann Kile from Lime Ridge, Pennsylvania.

Permanent Address: 307 Woodlawn Drive
Fairborn, Ohio 45345

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1990	Master's Thesis

4. TITLE AND SUBTITLE
Developing Computer Literacy Among Incoming Air Force Institute of Technology Graduate Students

5. FUNDING NUMBERS

6. AUTHOR(S)
Stephen M. Heaps, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GSM/LSO/90S-14

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This research developed an introductory level, applications software oriented, computer course for the School of Systems and Logistics, Air Force Institute of Technology (AFIT). The course is used to introduce new students to the applications software used in the graduate program. The research was designed to answer four basic research questions:

1. What computer skills are necessary for minimum computer literacy at AFIT?
2. Which application software packages are used in the AFIT graduate program?
3. Given the course-time available to teach QMGT 290, what should be taught to maximize the literacy of the incoming students?; and
4. What teaching method(s) is/are best to teach computer applications?

The author developed the revised QMGT 290 using a curriculum building model extracted from professional literature. The revised course contained a combination of lecture, hands-on, and take-home problems and exercises. The objective of the revised QMGT 290 course was to ensure a minimum competency level among all new students.

Over 75% of the students completing the revised course agreed that the course met its primary objective. Comments were solicited from both the instructors and the students to improve the course in the future.

Keywords:

14. SUBJECT TERMS
computers, micro-computers, computer education, computer training, computer literacy, Theses, Education

15. NUMBER OF PAGES
149

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL